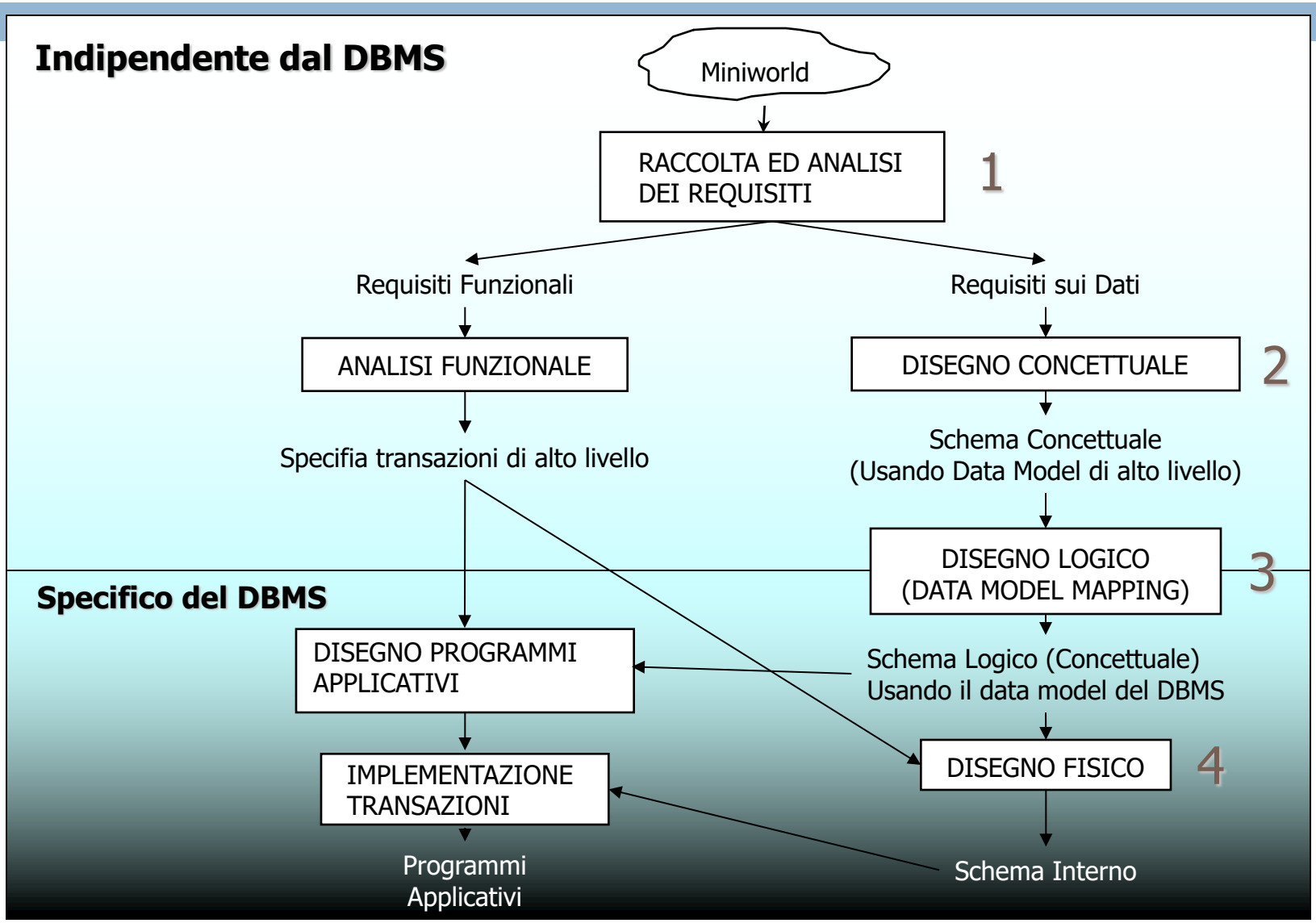
BASI DI DATI 2

DIPENDENZE FUNZIONALI E NORMALIZZAZIONE DI DB RELAZIONALI

a.a 2017/2018

Prof.ssa G. Tortora

Le fasi di progettazione di un Database



Qualità di schemi relazionali

- Non sempre dalla fase di progettazione si ottiene uno schema privo di difetti:
  - Possono sorgere dei problemi nella fase di mapping da **ER** a **Relazionale**.
  - Si può ottenere uno schema non ottimale progettando direttamente uno schema logico saltando la fase di analisi concettuale.

- Tali difetti possono portare ad anomalie nella base di dati.

Qualità di schemi relazionali (2)

- Come misurare la “bontà” di uno schema di relazione?
- La discussione è a due livelli:
  - livello logico
    - riferito al significato dello schema di relazione e degli attributi;
    - la bontà, a questo livello, aiuta l’utente a capire chiaramente il significato dei dati ed a formulare query correttamente;
  - livello di memorizzazione e manipolazione
    - Riguarda come le tuple sono memorizzate e aggiornate (solo per le relazioni base);

Misure informali di qualità

- Esistono delle misure informali di qualità per il disegno di schemi di relazione:
  1. Semantica degli attributi.
  2. Riduzione dei valori ridondanti nelle tuple.
  3. Riduzione dei valori **null** nelle tuple.
  4. Non consentire tuple spurie.

- Tali misure non sempre sono indipendenti.

Outline

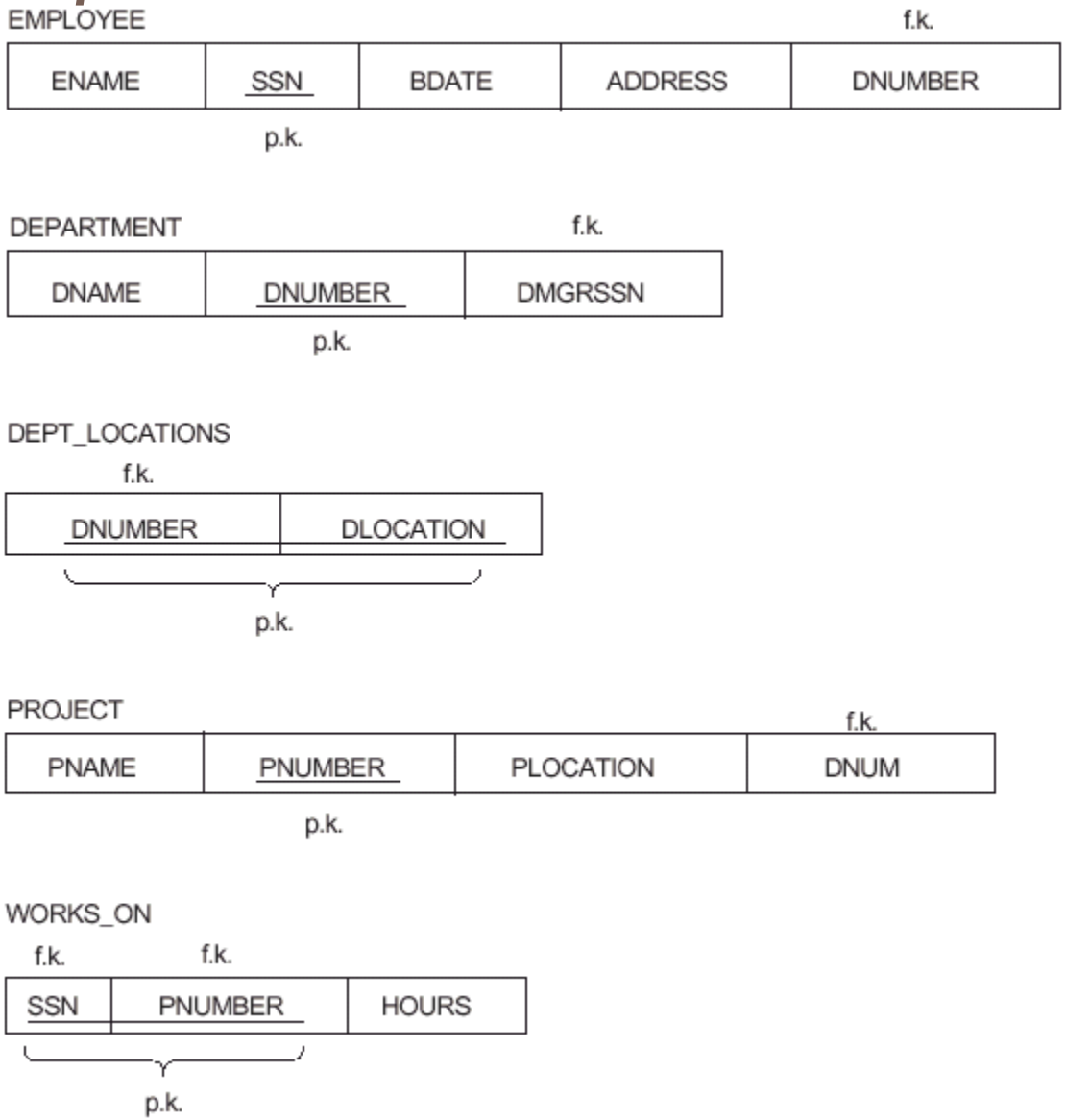
1. Semantica degli attributi.
2. Riduzione dei valori ridondanti nelle tuple.
3. Riduzione dei valori **null** nelle tuple.
4. Non consentire tuple spurie.

1) Semantica degli attributi di una relazione

- Quando si raggruppano gli attributi in uno schema di relazione, assumiamo che essi abbiano associato un significato.

- Il significato, o semantica, specifica come interpretare i valori degli attributi di una relazione:
  - Più è semplice spiegare la semantica della relazione, migliore è il disegno dello schema di relazione.

1) Semantica degli attributi di una relazione: *Esempio*



Versione semplificata dello schema del database Company

1) Semantica degli attributi di una relazione: *Esempio* (2)

EMPLOYEE

ENAME	SSN	BDATE	ADDRESS	DNUMBER
Smith,John B.	123456789	1965-01-09	731 Fondren,Houston,TX	5
Wong,Franklin T.	333445555	1955-12-08	638 Voss,Houston,TX	5
Zelaya,Alicia J.	999887777	1968-07-19	3321 Castle,Spring,TX	4
Wallace,Jennifer S.	987654321	1941-06-20	291 Berry,Bellaire,TX	4
Narayan,Ramesh K.	666884444	1962-09-15	975 Fire Oak,Humble,TX	5
English,Joyce A.	453453453	1972-07-31	5631 Rice,Houston,TX	5
Jabbar,Ahmad V.	987987987	1969-03-29	980 Dallas,Houston,TX	4
Borg,James E.	888665555	1937-11-10	450 Stone,Houston,TX	1

DEPARTMENT

DNAME	DNUMBER	DMGRSSN
Research	5	333445555
Administration	4	987654321
Headquarters	1	888665555

DEPT_LOCATIONS

DNUMBER	DLOCATION
1	Houston
4	Stafford
5	Bellaire
5	Sugarland
5	Houston

WORKS_ON

SSN	PNUMBER	HOURS
123456789	1	32.5
123456789	2	7.5
666884444	3	40.0
453453453	1	20.0
453453453	2	20.0
333445555	2	10.0
333445555	3	10.0
333445555	10	10.0
333445555	20	10.0
999887777	30	30.0
999887777	10	10.0
987987987	10	35.0
987987987	30	5.0
987654321	30	20.0
987654321	20	15.0
888665555	20	null

PROJECT

PNAME	PNUMBER	PLOCATION	DNUM
ProductX	1	Bellaire	5
ProductY	2	Sugarland	5
ProductZ	3	Houston	5
Computerization	10	Stafford	4
Reorganization	20	Houston	1
Newbenefits	30	Stafford	4

Esempio di relazioni dello schema del db Company

1) Semantica degli attributi di una relazione: *Esempio* (3)

- Significato dello schema del database:
  - ENAME, SSN, BDATE, ADDRESS → dati dell'impiegato.
  - DNUMBER → dipartimento per cui lavora.
  - DNUMBER (foreign key) → relazione implicita tra EMPLOYEE e DEPARTMENT.
  - DEPARTMENT → un'entità dipartimento.
  - PROJECT → una entità progetto.
  - f.k. DMGRSSN (di DEPARTMENT) → correla il dipartimento all'impiegato che è suo manager.

1) Semantica degli attributi di una relazione: *Esempio (4)*

- f.k. DNUM (di PROJECT) → correla un progetto al suo dipartimento di controllo.
- Ogni tupla in DEPT_LOCATIONS contiene un numero di dipartimento (DNUMBER) e una delle locazioni del dipartimento (DLOCATIONS).
- Ogni tupla in WORKS_ON contiene l'SSN di un impiegato, il numero di progetto PNUMBER di uno dei progetti su cui lavora l'impiegato, e il numero di ore settimanali HOURS.

1) Semantica degli attributi di una relazione: *Esempio (5)*

- DEPT_LOCATIONS → rappresenta un attributo multivalued di DEPARTMENT.
- WORKS_ON → rappresenta una relazione di cardinalità M:N tra EMPLOYEE e PROJECT.

- Tutte le relazioni quindi possono essere considerate *"buone"* dal punto di vista di avere una semantica chiara.

Linea guida 1

- Disegnare uno schema di relazione del quale sia facile spiegarne il significato.

- Non combinare attributi da tipi di entità e tipi di relazioni differenti in una singola relazione.
  - Intuitivamente: se uno schema di relazione corrisponde ad un tipo di entità o ad un tipo di relazione, il significato tende ad essere chiaro.

Linea guida 1: *Esempio*

- Considerando le due relazioni:

EMP_DEPT

ENAME	SSN	BDATE	ADDRESS	DNUMBER	DNAME	DMGRSSN

EMP_PROJ

SSN	PNUMBER	HOURS	ENAME	PNAME	PLOCATION

  - EMPT_DEPT → ogni entità rappresenta un impiegato ma include informazioni sul dipartimento in cui lavora.
  - EMP_PROJ → ogni tupla correla un impiegato a un progetto ma richiede anche il nome dell'impiegato ENAME e il nome e la locazione del progetto, PNAME e PLOCATION.

- Tali schemi di relazione hanno anch'essi una semantica chiara, però entrambe contravvengono la linea guida 1, contenendo attributi di entità distinte.

Outline

1. Semantica degli attributi.
2. Riduzione dei valori ridondanti nelle tuple.
3. Riduzione dei valori **null** nelle tuple.
4. Non consentire tuple spurie.

2) Riduzione dei valori ridondanti nelle tuple

- Un obiettivo del disegno dello schema è di minimizzare lo spazio di memoria occupato dalle relazioni base.

2) Riduzione dei valori ridondanti nelle tuple: *Esempio*

EMPLOYEE

ENAME	SSN	BDATE	ADDRESS	DNUMBER
Smith,John B.	123456789	1965-01-09	731 Fondren,Houston,TX	5
Wong,Franklin T.	333445555	1955-12-08	638 Voss,Houston,TX	5
Zelaya,Alicia J.	999887777	1968-07-19	3321 Castle,Spring,TX	4
Wallace,Jennifer S.	987654321	1941-06-20	291 Berry,Bellaire,TX	4
Narayan,Ramesh K.	666884444	1962-09-15	975 Fire Oak,Humble,TX	5
English,Joyce A.	453453453	1972-07-31	5631 Rice,Houston,TX	5
Jabbar,Ahmad V.	987987987	1969-03-29	980 Dallas,Houston,TX	4
Borg,James E.	888665555	1937-11-10	450 Stone,Houston,TX	1

DEPARTMENT

DNAME	DNUMBER	DMGRSSN
Research	5	333445555
Administration	4	987654321
Headquarters	1	888665555

DEPT_LOCATIONS

DNUMBER	DLOCATION
1	Houston
4	Stafford
5	Bellaire
5	Sugarland
5	Houston

WORKS_ON

SSN	PNUMBER	HOURS
123456789	1	32.5
123456789	2	7.5
666884444	3	40.0
453453453	1	20.0
453453453	2	20.0
333445555	2	10.0
333445555	3	10.0
333445555	10	10.0
333445555	20	10.0
999887777	30	30.0
999887777	10	10.0
987987987	10	35.0
987987987	30	5.0
987654321	30	20.0
987654321	20	15.0
888665555	20	null

PROJECT

PNAME	PNUMBER	PLOCATION	DNUM
ProductX	1	Bellaire	5
ProductY	2	Sugarland	5
ProductZ	3	Houston	5
Computerization	10	Stafford	4
Reorganization	20	Houston	1
Newbenefits	30	Stafford	4

Schema corretto

2) Riduzione dei valori ridondanti nelle tuple: *Esempio* (2)

EMP_DEPT

ENAME	SSN	BDATE	ADDRESS	DNUMBER	DNAME	DMGRSSN
Smith,John B.	123456789	1965-01-09	731 Fondren,Houston,TX	5	Research	333445555
Wong,Franklin T.	333445555	1955-12-08	638 Voss,Houston,TX	5	Research	333445555
Zelaya, Alicia J.	999887777	1968-07-19	3321 Castle,Spring,TX	4	Administration	987654321
Wallace,Jennifer S.	987654321	1941-06-20	291 Berry,Bellaire,TX	4	Administration	987654321
Narayan,Ramesh K.	666884444	1962-09-15	975 FireOak,Humble,TX	5	Research	333445555
English,Joyce A.	453453453	1972-07-31	5631 Rice,Houston,TX	5	Research	333445555
Jabbar,Ahmad V.	987987987	1969-03-29	980 Dallas,Houston,TX	4	Administration	987654321
Borg,James E.	888665555	1937-11-10	450 Stone,Houston,TX	1	Headquarters	888665555

EMP_PROJ

SSN	PNUMBER	HOURS	ENAME	PNAME	PLOCATION
123456789	1	32.5	Smith,John B.	ProductX	Bellaire
123456789	2	7.5	Smith,John B.	ProductY	Sugarland
666884444	3	40.0	Narayan,Ramesh K.	ProductZ	Houston
453453453	1	20.0	English,Joyce A.	ProductX	Bellaire
453453453	2	20.0	English,Joyce A.	ProductY	Sugarland
333445555	2	10.0	Wong,Franklin T.	ProductY	Sugarland
333445555	3	10.0	Wong,Franklin T.	ProductZ	Houston
333445555	10	10.0	Wong,Franklin T.	Computerization	Stafford
333445555	20	10.0	Wong,Franklin T.	Reorganization	Houston
999887777	30	30.0	Zelaya,Alicia J.	Newbenefits	Stafford
999887777	10	10.0	Zelaya,Alicia J.	Computerization	Stafford
987987987	10	35.0	Jabbar,Ahmad V.	Computerization	Stafford
987987987	30	5.0	Jabbar,Ahmad V.	Newbenefits	Stafford
987654321	30	20.0	Wallace,Jennifer S.	Newbenefits	Stafford
987654321	20	15.0	Wallace,Jennifer S.	Reorganization	Houston
888665555	20	null	Borg,James E.	Reorganization	Houston

Schema errato

2) Riduzione dei valori ridondanti nelle tuple: *Esempio* (3)

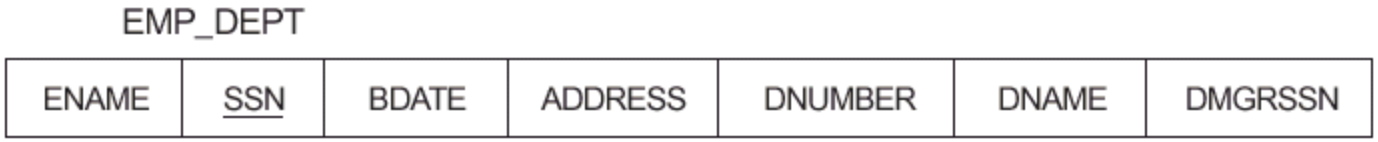
- Oltre all'occupazione di più spazio, le relazioni EMP_DEPT e EMP_PROJ usate come relazioni base, presentano anche il problema delle *"anomalie di aggiornamento"* (update anomalies)…

Update Anomalies

- Le "*update anomalies*" sono delle anomalie che possono sorgere nel gestire i dati di un database relazionale non correttamente progettato.
- Si dividono in:
  - *Insertion Anomalies*
  - *Deletion Anomalies*
  - *Modification Anomalies*

Insertion Anomalies

- Possono essere di due tipi, mostrati riferendosi allo schema EMP_DEPT:



1. Per inserire una nuova tupla impiegato, dobbiamo anche richiedere i valori degli attributi del dipartimento per cui lavora.

   Inoltre tali valori devono essere consistenti.

   Esempio: inserendo un nuovo impiegato che lavora nel dipartimento 5, dobbiamo inserire i valori degli attributi DNAME e DMGRSSN consistenti con tali attributi di altre tuple in EMP_DEPT per il dipartimento 5.

Insertion Anomalies (2)

EMP_DEPT

ENAME	SSN	BDATE	ADDRESS	DNUMBER	DNAME	DMGRSSN

2. È difficile inserire un nuovo dipartimento che non ha ancora impiegati nella relazione EMP_DEPT.

   L'unico modo è di inserirlo ponendo a **null** gli attributi per l'impiegato, ma ciò crea un problema poiché SSN è la chiave primaria.
   Inoltre inserendo il primo impiegato per quel dipartimento non servirebbe più la tupla con valori **null**.

Deletion Anomalies

- Questo problema è correlato al secondo problema delle insertion anomalies.

- Se cancelliamo da EMP_DEPT una tupla impiegato che rappresenta l'ultimo impiegato che lavora per un particolare dipartimento, l'informazione sul dipartimento è persa dal database.

Modification Anomalies

- Se cambiamo il valore di uno degli attributi di un particolare dipartimento, dobbiamo aggiornare tutte le tuple di tutti gli impiegati che lavorano per quel dipartimento.
  Altrimenti il database diventa inconsistente.

Linea guida 2

- Disegnare gli schemi di relazione di base in modo che non possano accadere insertion, deletion o modification anomalies.
  - Se qualcuna è presente, annotare ciò chiaramente in modo che i programmi che aggiornano il database operino correttamente.

- Talvolta le linee guida possono essere violate per scopi di efficienza. Una soluzione migliore potrebbe comunque essere quella di definire delle viste.

Outline

1. Semantica degli attributi.
2. Riduzione dei valori ridondanti nelle tuple.
3. Riduzione dei valori **null** nelle tuple.
4. Non consentire tuple spurie.

3) Riduzione dei valori null nelle tuple

- In alcuni disegni di schemi possiamo raggruppare molti attributi in una relazione grossa. Se molti degli attributi non applicano a tutte le tuple possiamo avere molti valori **null**.

- Oltre allo spreco di spazio, ciò crea problemi con le funzioni di aggregazione COUNT e SUM.

Interpretazioni di null

- Il valore **null** può avere diverse interpretazioni:
  - L'attributo non si applica a questa tupla.
  - Il valore dell'attributo per questa tupla non è noto.
  - Il valore dell'attributo è noto ma assente, cioè non è stato ancora registrato.

Linea guida 3

- Evitare (*per quanto possibile*) di porre attributi in una relazione base i cui valori possono essere **null**.

- Se i valori **null** sono inevitabili, assicurarsi che essi ricorrano in casi eccezionali e non per la maggioranza delle tuple.

Linea guida 3: *Esempio*

- Se solo il 10% degli impiegati ha un ufficio individuale, è poco giustificato include l'attributo OFFICE_NUMBER nella relazione EMPLOYEE;

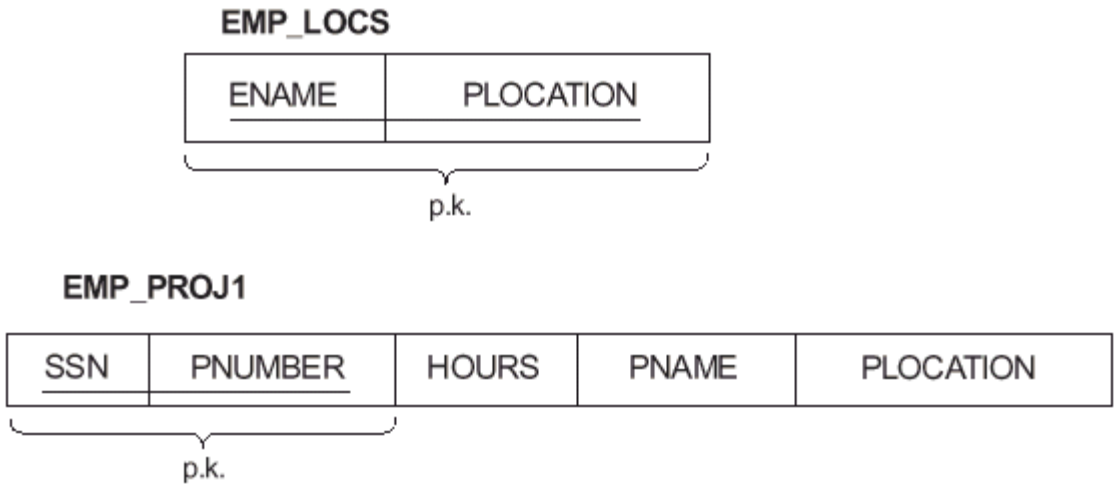
  piuttosto creare la relazione:
  EMP_OFFICES (ESSN,OFFICE_NUMBER)

Outline

1. Semantica degli attributi.
2. Riduzione dei valori ridondanti nelle tuple.
3. Riduzione dei valori **null** nelle tuple.
4. Non consentire tuple spurie.

Tuple spurie

- Consideriamo i due schemi, alternativi a EMP_PROJ:



- Le tuple rappresentano:
  - EMP_LOCS: l'impiegato di nome ENAME lavora per qualche progetto la cui locazione è PLOCATION.
  - EMP_PROJ1: l'impiegato con codice SSN lavora HOURS ore alla settimana sul progetto avente numero PNUMBER, nome PNAME e locazione PLOCATION.

Tuple spurie (2)

- La suddivisione appena vista non corrisponde ad un buon disegno di db in quanto, volendo riottenere le informazioni di EMP_PROJ mediante un natural join sui due schemi, si ottengono tuple spurie, rappresentanti informazioni errate.

Tuple spurie: *Esempio*

EMP_LOCS

ENAME	PLOCATION
Smith, John B.	Bellaire
Smith, John B.	Sugarland
Narayan, Ramesh K.	Houston
English, Joyce A.	Bellaire
English, Joyce A.	Sugarland
Wong, Franklin T.	Sugarland
Wong, Franklin T.	Houston
Wong, Franklin T.	Stafford

EMP_PROJ1

SSN	PNUMBER	HOURS	PNAME	PLOCATION
123456789	1	32.5	Product X	Bellaire
123456789	2	7.5	Product Y	Sugarland
666884444	3	40.0	Product Z	Houston
453453453	1	20.0	Product X	Bellaire
453453453	2	20.0	Product Y	Sugarland
333445555	2	10.0	Product Y	Sugarland
333445555	3	10.0	Product Z	Houston
333445555	10	10.0	Computerization	Stafford
333445555	20	10.0	Reorganization	Houston

I valori presenti nelle due relazioni

Tuple spurie: *Esempio* (2)

	SSN	PNUMBER	HOURS	PNAME	PLOCATION	
	123456789	1	32.5	ProductX	Bellaire	Smith,John B.
*	123456789	1	32.5	ProductX	Bellaire	English,Joyce A.
	123456789	2	7.5	ProductY	Sugarland	Smith,John B.
*	123456789	2	7.5	ProductY	Sugarland	English,Joyce A.
*	123456789	2	7.5	ProductY	Sugarland	Wong,Franklin T.
	666884444	3	40.0	ProductZ	Houston	Narayan,Ramesh K.
*	666884444	3	40.0	ProductZ	Houston	Wong,Franklin T.
*	453453453	1	20.0	ProductX	Bellaire	Smith,John B.
	453453453	1	20.0	ProductX	Bellaire	English,Joyce A.
*	453453453	2	20.0	ProductY	Sugarland	Smith,John B.
	453453453	2	20.0	ProductY	Sugarland	English,Joyce A.
*	453453453	2	20.0	ProductY	Sugarland	Wong,Franklin T.
*	333445555	2	10.0	ProductY	Sugarland	Smith,John B.
*	333445555	2	10.0	ProductY	Sugarland	English,Joyce A.
	333445555	2	10.0	ProductY	Sugarland	Wong,Franklin T.
*	333445555	3	10.0	ProductZ	Houston	Narayan,Ramesh K.
	333445555	3	10.0	ProductZ	Houston	Wong,Franklin T.
	333445555	10	10.0	Computerization	Stafford	Wong,Franklin T.
*	333445555	20	10.0	Reorganization	Houston	Narayan,Ramesh K.
	333445555	20	10.0	Reorganization	Houston	Wong,Franklin T.

Risultato del natural join:

le tuple spurie sono contrassegnate da un ''*

Motivo delle tuple spurie

- Tale problema sorge perché PLOCATION è l'attributo che correla EMP_LOCS e EMP_PROJ1, ma non è né chiave primaria, né chiave esterna in nessuna delle due relazioni.

Linea guida 4

- Progettare gli schemi di relazione in modo da poter effettuare JOIN con condizioni di uguaglianza su attributi che sono o chiave primaria o chiave esterna, in modo da non generare tuple spurie.

Sommario delle linee guida

- Abbiamo mostrato in modo informale che una cattiva progettazione dello schema di un db può portare ad una serie di problemi:
  - Anomalie che implicano un maggiore sforzo nell'inserimento e nella modifica di una relazione, e che possono portare a perdite accidentali di dati durante la cancellazione.
  - Spreco di spazio a causa dei valori **null**.
  - Generazione di tuple spurie o non valide durante operazioni di join.

Dipendenze Funzionali

Qualità di schemi relazionali

- Dopo una visione informale, vediamo delle teorie e dei concetti formali per descrivere la *"bontà"* dei singoli schemi di relazione con maggiore precisione, usando le *"dipendenze funzionali"* e le *"forme normali"*.

Dipendenza Funzionale

- Una dipendenza funzionale (FD) è un vincolo tra due insiemi di attributi del database.

- Supponiamo che lo schema di db relazionale abbia n attributi $A_1, A_2, \ldots, A_n$ e che l'intero database sia descritto da uno schema di relazione universale

$$R = \{A_1, A_2, \ldots, A_n\}.$$

- Una dipendenza funzionale, denotata da X→Y, tra due insiemi di attributi X e Y che sono sottoinsiemi di R, specifica un vincolo sulle possibili tuple che possono formare una istanza di relazione r di R.

Dipendenza Funzionale (2)

- Il vincolo stabilisce che se X → Y, allora
  ∀ t_1, t_2 in r tali che

  $$t_1[X] = t_2[X],$$

  deve valere

  $$t_1[Y] = t_2[Y].$$

  - Ciò significa che i valori della componente Y di una tupla di r dipendono da (o sono determinati da) i valori della componente X.

Dipendenza Funzionale (3)

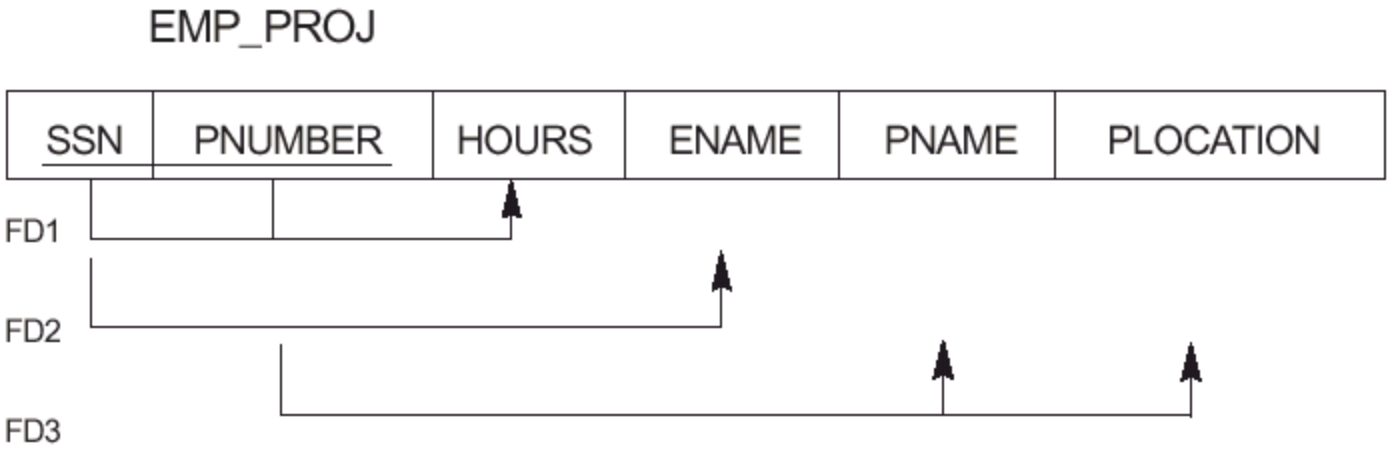
- Alternativamente, i valori della componente X di una tupla determinano univocamente (o funzionalmente) i valori della componente Y.
- Cioè esiste una dipendenza funzionale da X a Y:
  - Y è funzionalmente dipendente da X
  - X è la parte sinistra della FD
  - Y è la parte destra della FD

Dipendenza Funzionale (4)

- Si noti che:
  - Se un vincolo su R stabilisce che non può esistere più di una tupla con un dato valore per X in ogni istanza di relazione r(R): *"cioè X è una chiave candidata di R"*; ciò implica che X → Y per ogni sottoinsieme Y di attributi di R.
  - Il fatto che X → Y in R non implica nulla circa Y → X in R.

Dipendenza Funzionale: *Esempio*

- La dipendenza funzionale è una proprietà della semantica degli attributi.



  - FD3: PNUMBER → {PNAME, PLOCATION}
  - FD2: SSN → ENAME
  - FD1: {SSN, PNUMBER} → HOURS

Dipendenza funzionale dalla semantica degli attributi

- Una dipendenza funzionale è una proprietà dello schema di relazione (intenzione) R e non di un particolare stato di relazione (estensione legale r di R).
  - (Le estensioni legali o stati di relazione legali sono delle estensioni di relazione r(R) che soddisfano i vincoli di FD).

- Una FD non può essere inferita automaticamente da un'estensione di relazione r, ma deve essere definita esplicitamente da chi conosce la semantica degli attributi.

Dipendenza Funzionale: *Esempio*

TEACH

TEACHER	COURSE	TEXT
Smith	Data Structures	Bartram
Smith	Data Management	Al-Nour
Hall	Compilers	Hoffman
Brown	Data Structures	Augenthaler

- Si potrebbe pensare che Teacher determina funzionalmente Course e che Course determina funzionalmente Text.

  Ci sono però delle tuple che non seguono queste FD, e quindi giungiamo alla conclusione che:
  - TEACHER ~~→~~ COURSE
  - COURSE ~~→~~ TEXT

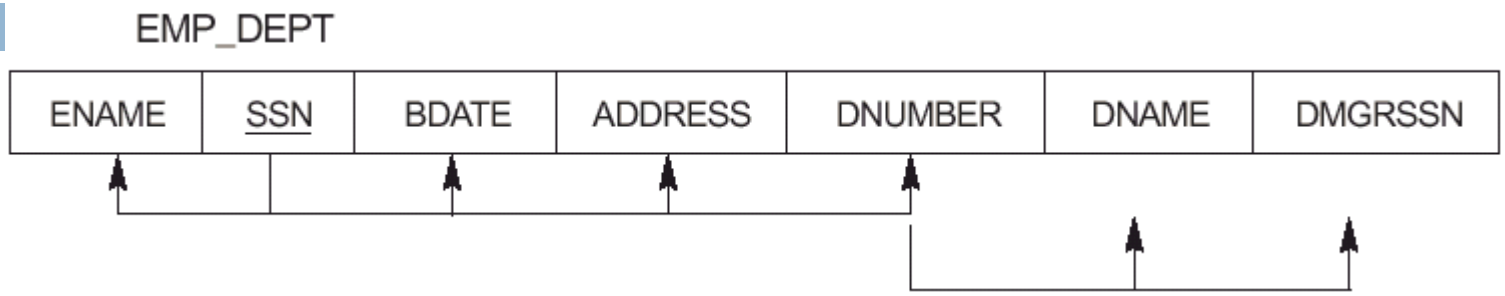
Regole di inferenza per le FD

- Sia F l'insieme delle dipendenze funzionali di uno schema di relazione R. Il disegnatore dello schema specifica le dipendenze funzionali semanticamente ovvie F.

- In tutte le istanze di relazione legali, però, valgono numerose altre dipendenze funzionali: queste possono essere inferite (o dedotte) da F.

- L'insieme di tali dipendenze è detto chiusura di F, ed è denotato da F^+.

Regole di inferenza: *Esempio*



- F = {
  - SSN → { ENAME, BDATE, ADDRESS, DNUMBER},
  - DNUMBER → {DNAME,DMGRSSN}
    }
- Possiamo inferire:
  - SSN → {DNAME,DMGRSSN}
  - SSN → SSN
  - DNUMBER → DNAME
  - …

Regole di inferenza per le FD (2)

- Una FD X→Y è inferita da un insieme di dipendenze F su R se X→Y vale in ogni stato di relazione r che è estensione legale di R.

- F ⊨ X→Y denota che la FD X→Y è inferita da F.

- Notazioni:
  - {X,Y} → Z è abbreviato in XY→Z
  - {X,Y,Z} → {U,V} è abbreviato in XYZ → UV

Regole di inferenza fondamentali

- Regole di inferenza per le dipendenze funzionali:
- (IR1) (Reflexive rule)
  - Se $X \supseteq Y$ allora $X \rightarrow Y$
- (IR2) (Augmentation rule)
  - $X \rightarrow Y \models XZ \rightarrow YZ$
- (IR3) (Transitive rule)
  - $\{X \rightarrow Y , Y \rightarrow Z\} \models X \rightarrow Z$

Regole di inferenza fondamentali (2)

- (IR4) (Decomposition or Projective rule)
  - $\{X \rightarrow YZ\} \models X \rightarrow Z$
- (IR5) (Union (or additive) rule)
  - $\{X \rightarrow Y, X \rightarrow Z\} \models X \rightarrow YZ$
- (IR6) (Pseudotransitive rule)
  - $\{X \rightarrow Y, WY \rightarrow Z\} \models WX \rightarrow Z$

Dimostrazione di IR1

Se $X \supseteq Y$ allora $X \rightarrow Y$

- La tesi $X \rightarrow Y$ significa che

  $\forall\ t_1, t_2, \quad t_1[X] = t_2[X] \ \Rightarrow\ t_1[Y] = t_2[Y]$
- Sia $X \supseteq Y$ e supponiamo che esistano tuple t_1, t_2 in qualche istanza r di R tale che $t_1[X] = t_2[X]$. Ma allora $t_1[Y] = t_2[Y]$ poiché $X \supseteq Y$.

  Quindi $X \rightarrow Y$.

Dimostrazione di IR2

$$\{X \rightarrow Y\} \models XZ \rightarrow YZ$$

- Per assurdo assumiamo che $X \rightarrow Y$ vale ma $XZ \rightarrow YZ$ non vale.
- Allora devono esistere t_1 e t_2 in r tali che

  (1) $t_1[X] = t_2[X]$, (2) $t_1[Y] = t_2[Y]$,

  (3) $t_1[XZ] = t_2[XZ]$ e (4) $t_1[YZ] \neq t_2[YZ]$.
  Ma ciò è falso perché:

  $t_1[X] = t_2[X]$ e $t_1[XZ] = t_2[XZ] \Rightarrow t_1[Z] = t_2[Z]$
  Inoltre

  $t_1[Y] = t_2[Y]$ e $t_1[Z] = t_2[Z] \Rightarrow t_1[YZ] = t_2[YZ]$ che contraddice l'assunto.

Dimostrazione di IR3

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

- Assumiamo che $X \rightarrow Y$ e $Y \rightarrow Z$ valgono in una relazione r.
- Allora $\forall\, t_1, t_2$ in r: $t_1[X] = t_2[X] \Rightarrow t_1[Y] = t_2[Y]$ e quindi poiché $Y \rightarrow Z$, $t_1[Y] = t_2[Y] \Rightarrow t_1[Z] = t_2[Z]$; quindi $X \rightarrow Z$ vale in r.

Dimostrazione di IR4

(IR4) - (IR6) possono essere provate usando (IR1) - (IR3)

(IR4): $\{X \rightarrow YZ\} \models X \rightarrow Y$

1. $X \rightarrow YZ$ (per ipotesi)
2. $YZ \rightarrow Y$ (da IR1 e dato che $YZ \supseteq Y$)
3. $X \rightarrow Y$ (da IR3, 1 e 2)

Dimostrazione di IR5

$$\{ X \rightarrow Y, X \rightarrow Z\} \models X \rightarrow YZ$$

1. $X \rightarrow Y$ (per ipotesi)
2. $X \rightarrow Z$ (per ipotesi)
3. $X \rightarrow XY$

   Poiché $X = XX$ allora $X \rightarrow XX$, e da 1. $X \rightarrow XY$
4. $XY \rightarrow YZ$ da 2. e usando IR2 (aumentando con Y)
5. $X \rightarrow YZ$ da IR3, 3. e 4.

Dimostrazione di IR6

$$\{ X \rightarrow Y, WY \rightarrow Z\} \models WX \rightarrow Z$$

1. $X \rightarrow Y$ (per ipotesi)
2. $WY \rightarrow Z$ (per ipotesi)
3. $WX \rightarrow WY$ da 1. e IR2
4. $WX \rightarrow Z$ da 3. e 2. con IR3.

Regole di inferenza di Armstrong

- È stato provato che (IR1) - (IR3) sono regole di inferenza corrette e complete (Armstrong 1974):
  - Corretto: Dato un insieme F di dipendenze funzionali su R ogni dipendenza inferita da F usando IR1 ÷ IR3 è vera in ogni stato che soddisfa le dipendenze in F.
  - Completo: la chiusura di F (l'insieme di tutte le possibili dipendenze inferibili da F) può essere determinata usando solo IR1 - IR3 (regole di inferenza di Armstrong).

Regole di inferenza nella progettazione dei db

- Tipicamente, il progettista del db prima specifica le dipendenze funzionali F a partire dalla semantica degli attributi e poi usa le regole di inferenza di Armstrong per inferire dipendenze funzionali aggiuntive.
- Metodo :
  1. determinare ogni insieme di attributi che appare come parte sinistra di qualche dipendenza funzionale in F.
  2. usare le regole di Armstrong per determinare l'insieme di tutti gli attributi dipendenti da X.

Algoritmo per il calcolo di X^+

$X^+ := X$;

repeat

  $oldX^+ := X^+$;

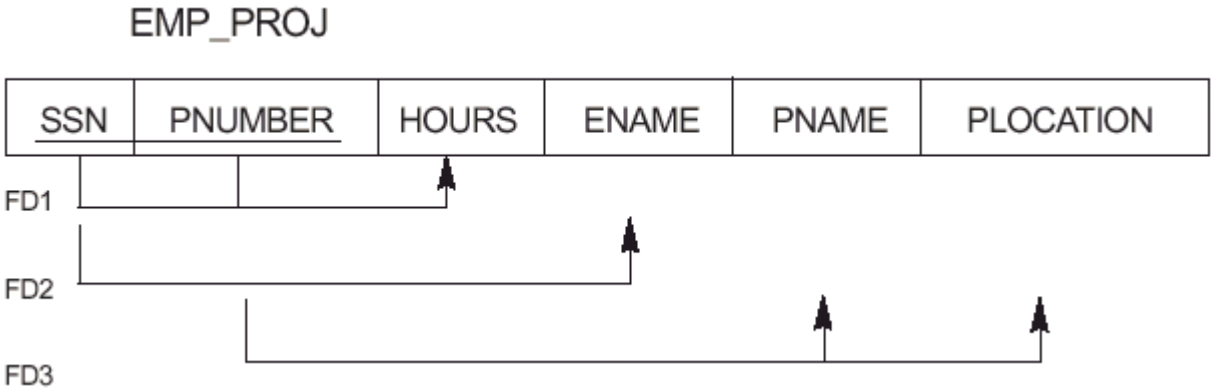
  for each functional dependency $Y \rightarrow Z$ **in** F **do**

    if $X^+ \supseteq Y$

    then $X^+ := X^+ \cup Z$;

until $(oldX^+ = X^+)$;

Algoritmo per il calcolo di X^+: *Esempio*



- Dalla semantica degli attributi si ha:
- F={ SSN → ENAME,
  PNUMBER → { PNAME, PLOCATION},
  {SSN, PNUMBER} → HOURS }
- Usando l'algoritmo calcoliamo:
  - $\{SSN\}^+ = \{SSN, ENAME\}$
  - $\{PNUMBER\}^+ = \{PNUMBER, PNAME, PLOCATION\}$
  - $\{SSN, PNUMBER\}^+ = \{SSN, PNUMBER, ENAME, PNAME, PLOCATIONS, HOURS\}$

Equivalenza di insiemi di dipendenze funzionali

- **Definizione:**
  - Un insieme F di dipendenze funzionali **copre** un altro insieme E di dipendenze funzionali, se ogni DF in E è presente anche in F^+, cioè se ogni dipendenza in E può essere inferita a partire da F.
- **Definizione:**
  - Due insiemi E ed F di dipendenze funzionali sono equivalenti se $E^+ = F^+$; ossia E è equivalente ad F se sussistono entrambe le condizioni: E **copre** F e F **copre** E.
- Si può determinare se F **copre** E calcolando X^+ *rispetto* a F per ogni DF X → Y in E, e quindi verificando se questo X^+ comprende gli attributi presenti in Y.

Forme Normali

Normalizzazione dei dati

- La normalizzazione dei dati può essere vista come un processo che consente di decomporre schemi di relazione non ottimali in schemi più piccoli, tali da garantire la mancanza di *“update anomalies”*.

Forme normali

- Le forme normali forniscono
  - Un ambito formale per analizzare schemi di relazione, basato sulle chiavi e sulle dipendenze funzionali tra attributi.
  - Una serie di test da condurre sugli schemi di relazione individuali:
    - se un test fallisce, la relazione che viola il test deve essere decomposta in relazioni che individualmente superano il test.

Forme normali (2)

- Le prime tre forme normali (1NF, 2NF, 3NF) e la Boyce-Codd (BCNF) sono definite considerando solo vincoli di dipendenza funzionale e di chiave.
- La 4NF considera la *dipendenza multivalued*.
- La 5NF considera la *join dependency*.

Definizioni

- Una superchiave di uno schema di relazione $R = \{A_1, A_2, \ldots, A_n\}$ è un insieme di attributi $S \subseteq R$ con la proprietà che non esistono due tuple t_1 e t_2 in qualche stato di relazione legale di r di R tali che $t_1[S] = t_2[S]$.
- Una chiave k è una superchiave con la proprietà aggiuntiva che la rimozione da k di un qualche attributo fa perdere a k la proprietà di superchiave.

Definizioni (2)

- Tutte le chiavi *"minimali"* sono dette chiavi candidate. Una tra esse viene scelta arbitrariamente ed è detta chiave primaria.
- Un attributo A di uno schema di relazione R è primo se e solo se fa parte di almeno una chiave di R.
  - In caso contrario A è detto non-primo.

Definizioni: *Esempio*

EMPLOYEE

ENAME	SSN	BDATE	ADDRESS	DNUMBER

- {SSN} è una chiave
- {SSN}, {SSN,ENAME}, {SSN, ENAME, BDATE} etc. sono superchiavi
- {SSN} è l'unica chiave candidata
- {SSN} è la chiave primaria.

Prima forma normale (1NF)

- La 1NF è stata definita per non consentire attributi multivalued, composti e loro combinazioni.
  - Ora fa parte della normale definizione di relazione del modello relazionale.

- Gli unici valori consentiti da 1NF sono valori atomici (o indivisibili).

Prima forma normale: *Esempio*

DEPARTMENT

DNAME	DNUMBER	DMGRSSN	DLOCATIONS
Research	5	333445555	{Bellaire, Sugarland, Houston}
Administration	4	987654321	{Stafford}
Headquarters	1	888665555	{Houston}

- L'idea è di rimuovere DLOCATIONS che viola la 1NF e porlo in una relazione separata insieme con la chiave primaria.
  La chiave primaria di questa relazione è la combinazione di {DNUMBER, DLOCATION}.

DEPARTMENT

DNAME	DNUMBER	DMGRSSN
Research	5	333445555
Administration	4	987654321
Headquarters	1	888665555

DEPT_LOCATIONS

DNUMBER	DLOCATION
1	Houston
4	Stafford
5	Bellaire
5	Sugarland
5	Houston

Prima forma normale: *Esempio* (2)

- Un secondo modo di normalizzare è di avere una tupla per ogni locazione; in tal caso la chiave primaria è {DNUMBER, DLOCATION}, ma si ha ridondanza tra le tuple.

DEPARTMENT

DNAME	DNUMBER	DMGRSSN	DLOCATION
Research	5	333445555	Bellaire
Research	5	333445555	Sugarland
Research	5	333445555	Houston
Administration	4	987654321	Stafford
Headquarters	1	888665555	Houston

Prima forma normale: *Esempio* (3)

- Un terzo modo consiste nello stabilire un massimo numero di valori per l'attributo DLOCATION – ad es. 3 – e rimpiazzare l'attributo con i 3 attributi atomici DLOCATION1, DLOCATION2 e DLOCATION3.

SVANTAGGIO: introduzione di valori nulli.

Prima forma normale: *Esempio 2*

- La 1NF proibisce anche attributi composti (relazioni annidate):

EMP_PROJ

SSN	ENAME	PROJS	
		PNUMBER	HOURS

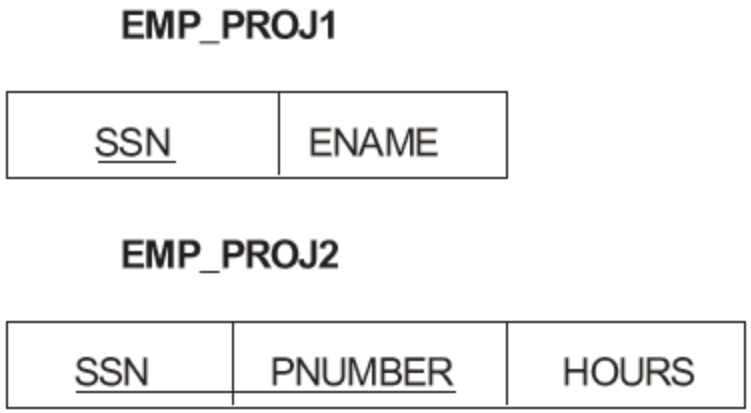
- SSN è chiave primaria.
- PNUMBER è chiave primaria parziale della relazione annidata.

EMP_PROJ

SSN	ENAME	PNUMBER	HOURS
123456789	Smith,John B.	1	32.5
		2	7.5
666884444	Narayan,Ramesh K.	3	40.0
453453453	English,Joyce A.	1	20.0
		2	20.0
333445555	Wong,Franklin T.	2	10.0
		3	10.0
		10	10.0
		20	10.0

Prima forma normale: *Esempio 2* (2)

- Per normalizzare:
  - rimuovere gli attributi della relazione annidata in una nuova relazione e propagare la chiave primaria in essa.



Seconda Forma Normale (2NF)

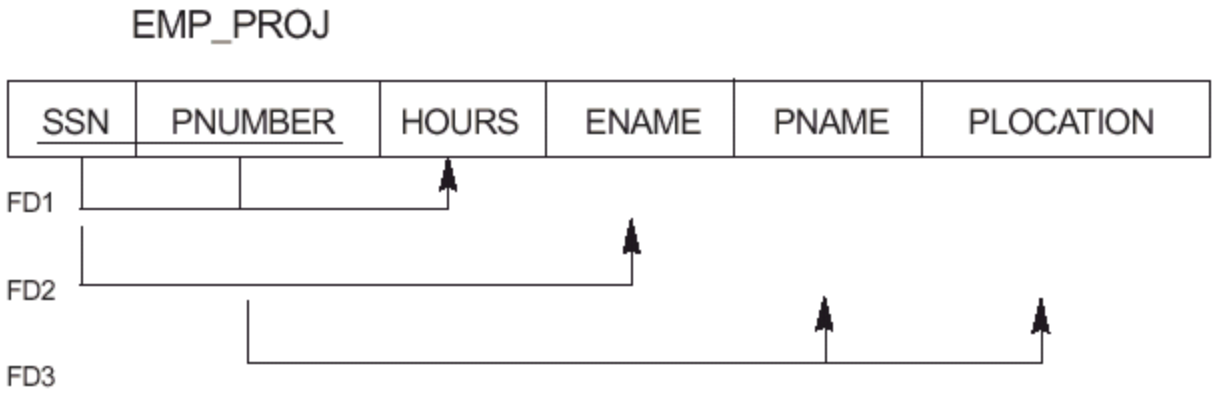
- È basata sul concetto di dipendenza funzionale piena.
- Definizione:
  - Una dipendenza funzionale X → Y è piena se la rimozione di qualche attributo A da X implica che la dipendenza non vale più, cioè :

    $\forall A \in X, (X - \{A\}) \not\rightarrow Y$

  - Una dipendenza funzionale X → Y è parziale se qualche attributo $A \in X$ può essere rimosso e la dipendenza vale ancora , cioè :

    $\forall A \in X, (X - \{A\}) \rightarrow Y$

Esempio dipendenze piene

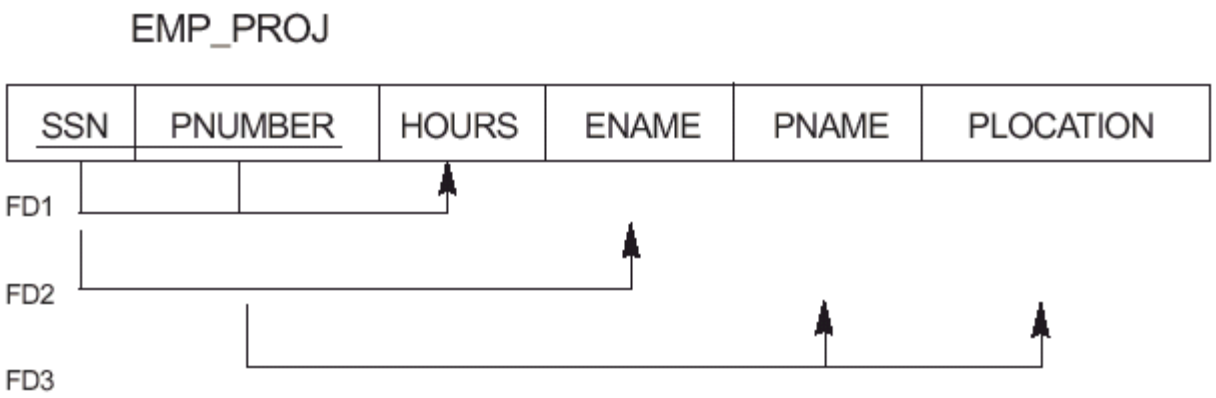


- {SSN, PNUMBER} → HOURS: piena infatti
  - SSN ↛ HOURS
  - PNUMBER ↛ HOURS
- {SSN, PNUMBER} → ENAME: parziale infatti
  - SSN → ENAME

Seconda Forma Normale (2NF)

- Uno schema di relazione R è in 2NF se:
  - R è in prima forma normale.
  - ogni attributo non-primo A in R è pienamente funzionalmente dipendente dalla chiave primaria di R.

Seconda Forma Normale: *Esempio*

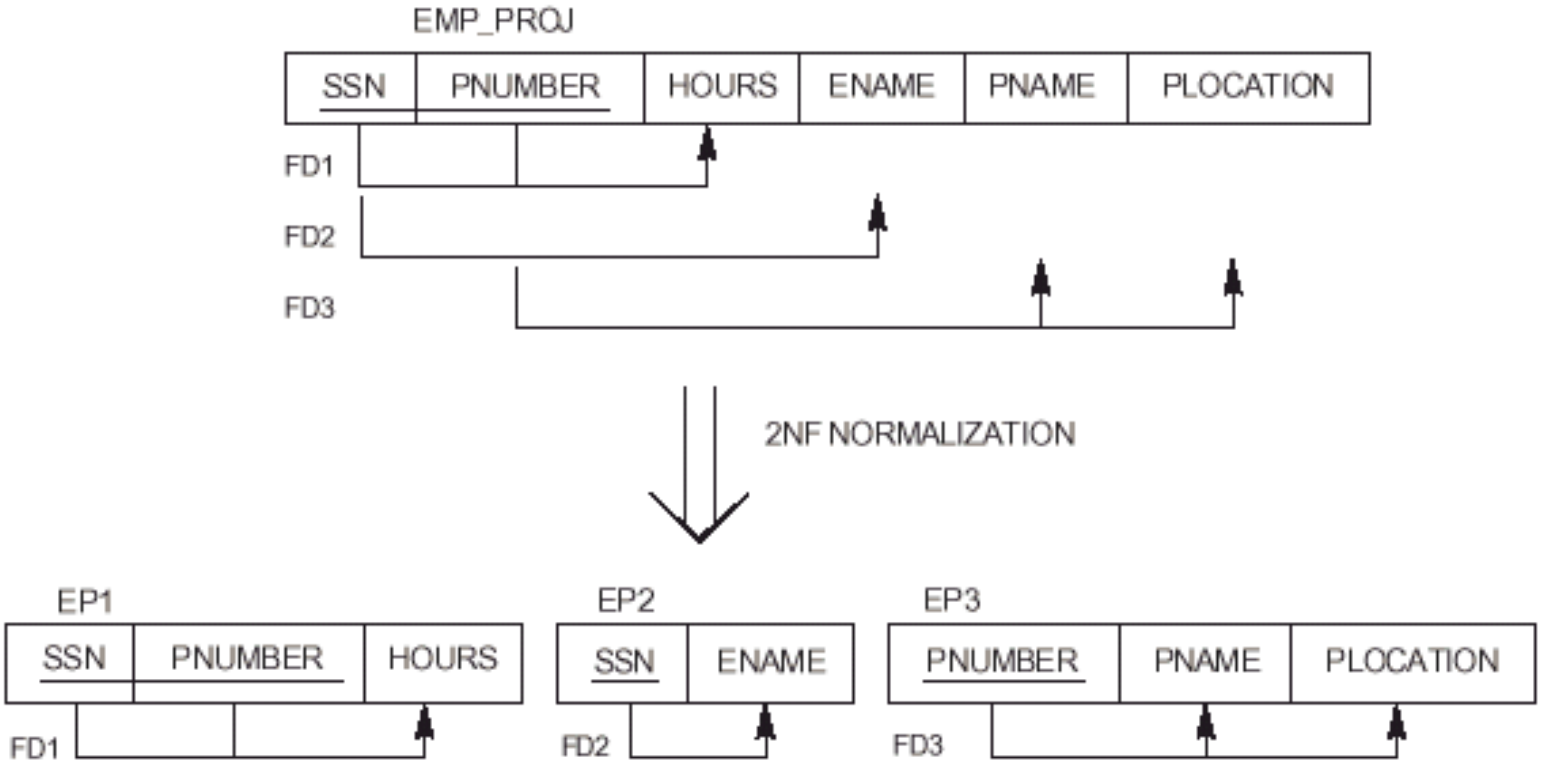


- EMP_PROJ è in 1NF ma non in 2NF.
- Infatti gli attributi non primi:
  - PNAME, PLOCATION violano 2NF in virtù di fd3.
  - ENAME viola 2NF in virtù di fd2.
- Le dipendenze funzionali fd2 e fd3 rendono ENAME, PNAME, PLOCATION parzialmente dipendenti dalla chiave primaria {SSN, PNUMBER}

Seconda Forma Normale

- Se uno schema di relazione non è in 2NF, può essere ulteriormente normalizzato in un certo numero di relazioni in 2NF, in cui gli attributi non primi sono associati solo con la parte di chiave primaria da cui sono funzionalmente dipendenti pienamente.

Esempio di 2NF



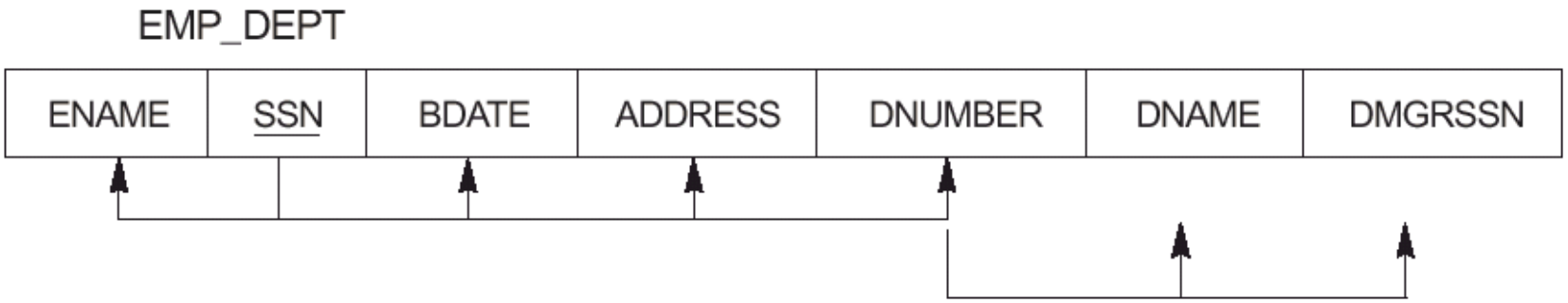
Normalizzazione in 2NF della relazione EMP_PROJ

Terza Forma Normale (3NF)

- È basata sul concetto di dipendenza transitiva.

- Una dipendenza funzionale X → Y in uno schema R è una dipendenza transitiva se esiste un insieme di attributi Z che non è sottoinsieme di alcuna chiave di R e valgono X → Z e Z → Y.

Terza Forma Normale: *Esempio*



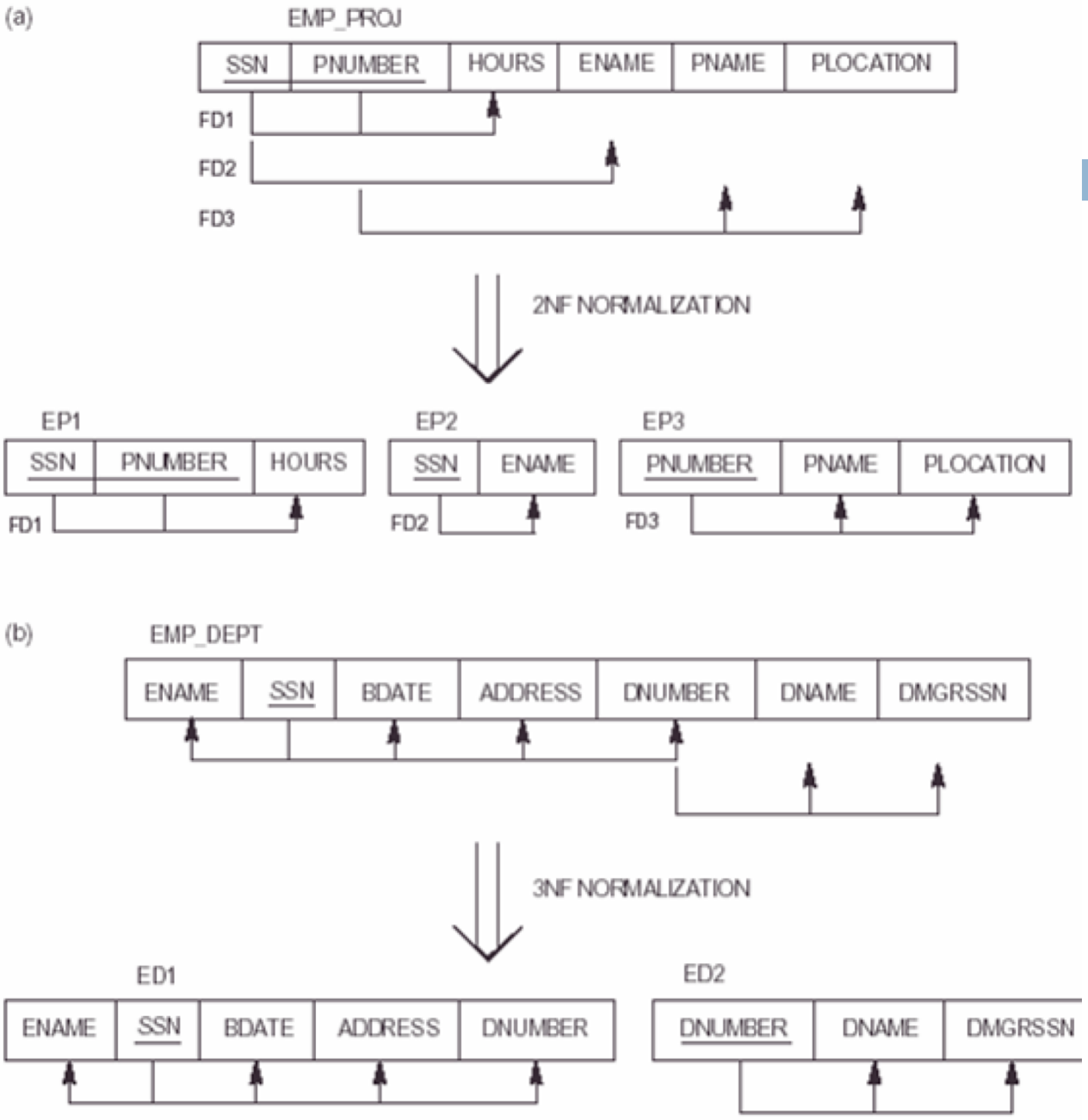
- SSN → DMGRSSN è transitiva attraverso DNUMBER infatti
  - SSN → DNUMBER
  - DNUMBER → DMGRSSN
  - DNUMBER non è sottoinsieme della chiave di EMP_DEPT

Terza Forma Normale

- Secondo le definizione di Codd:
  uno schema di relazione R è in 3NF se:
  - è in 2NF.
  - nessun attributo non-primo di R è transitivamente dipendente dalla chiave primaria.

Sommario

Forma normale	Test	Normalizzazione
1NF	La relazione non deve presentare attributi non atomici.	Forma una nuova relazione per ogni attributo non atomico.
2NF	La chiave primaria contiene attributi multipli e attributi non chiave sono funzionalmente dipendenti da parte della chiave primaria.	Crea una relazione per ogni chiave parziale con i suoi attributi dipendenti. (mantieni una relazione per la chiave primaria e gli attributi che dipendono funzionalmente da questa)
3NF	Non deve esistere una dipendenza transitiva tra un attributo non chiave e la chiave primaria.	Decomponi la relazione in un insieme di relazioni che includono gli attributi non chiave che funzionalmente determinano altri attributi non chiave.


(a)
EMP_PROJ
SSN
PNUMBER
HOURS
ENAME
PNAME
PLOCATION
FD1
FD2
FD3
2NF NORMALIZATION
EP1
SSN
PNUMBER
HOURS
FD1
EP2
SSN
ENAME
FD2
EP3
PNUMBER
PNAME
PLOCATION
FD3
(b)
EMP_DEPT
ENAME
SSN
BDATE
ADDRESS
DNUMBER
DNAME
DMGRSSN
3NF NORMALIZATION
ED1
ENAME
SSN
BDATE
ADDRESS
DNUMBER
ED2
DNUMBER
DNAME
DMGRSSN

Definizione generale di Seconda e Terza Forma Normale

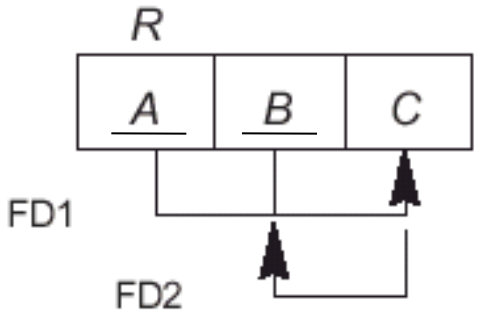
- La definizione generale di 2NF e 3NF non riguarda solo la chiave primaria ma tutte le chiavi candidate.
- Un attributo è primo se è parte di una qualsiasi chiave candidata.

- Uno schema di relazione R è in 2NF se ogni attributo non-primo A in R non è parzialmente dipendente da alcuna chiave di R.
- Uno schema di relazione R è in 3NF se, ogni volta che vale in R una dipendenza funzionale

  X → A, o
  - X è una superchiave di R, oppure
  - A è un attributo primo di R.

Forma Normale di Boyce-Codd (BCNF)

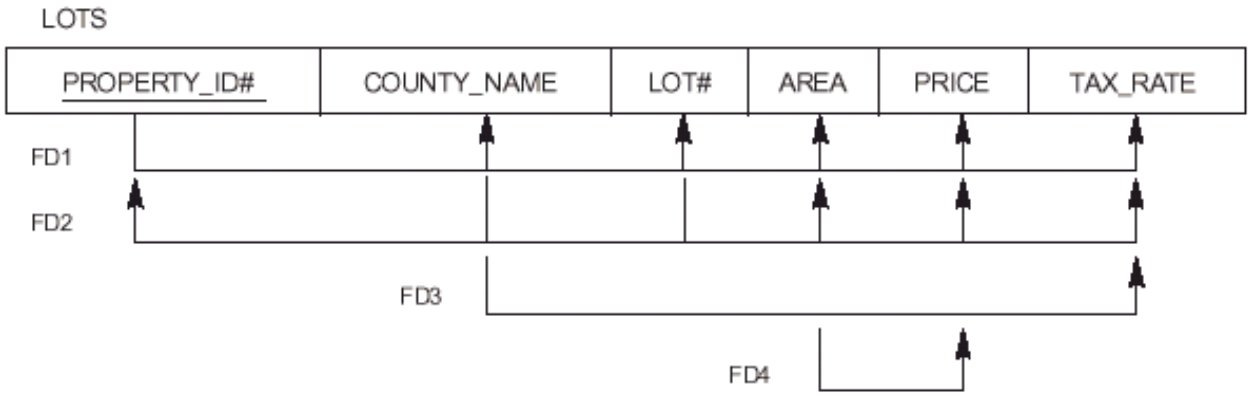
- Uno schema R è in BCNF se ogniqualvolta vale in R una dipendenza funzionale X → A, allora X è una superchiave di R.



Una relazione in 3NF, ma non in BCNF

Forme Normali: *Esempio*

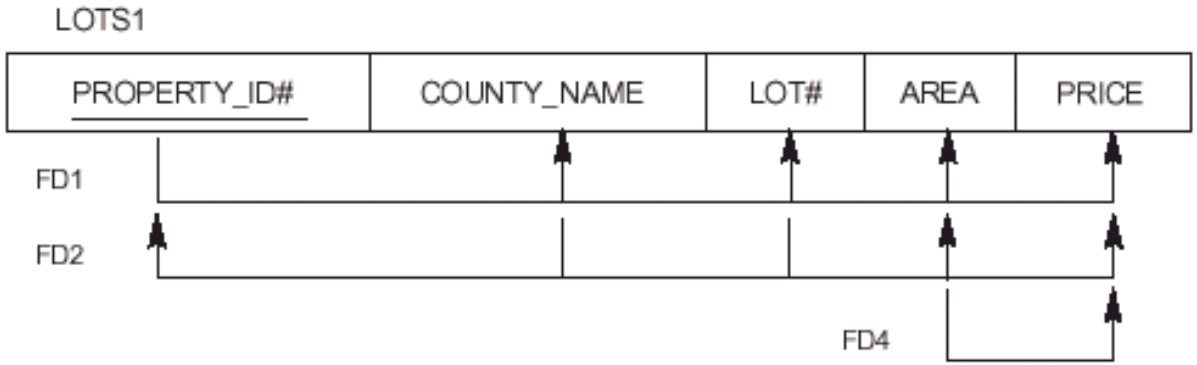
- Lo schema di relazione LOTS descrive lotti di terreno in vendita in varie contee di uno stato.

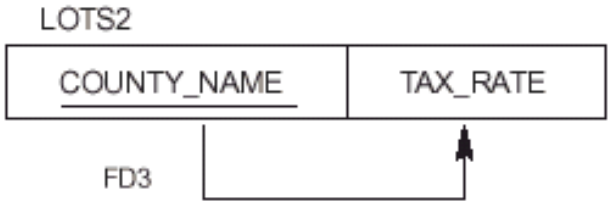


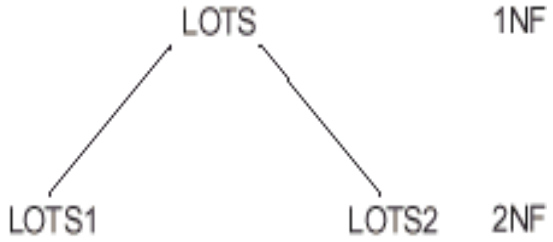
- Chiavi candidate: PROPERTY_ID# e {COUNTY_NAME, LOT#}
- Chiave primaria: PROPERTY_ID#
- Inoltre valgono:
  - FD3: COUNTY_NAME → TAX-RATE
  - FD4: AREA → PRICE

Forme Normali: *Esempio* (2)

- LOTS viola la definizione generale di 2NF perché TAX_RATE è parzialmente dipendente dalla chiave candidata {COUNTY_NAME, LOT#} in virtù di fd3.
- Per normalizzare decomponiamo LOTS in LOTS1 e LOTS2.







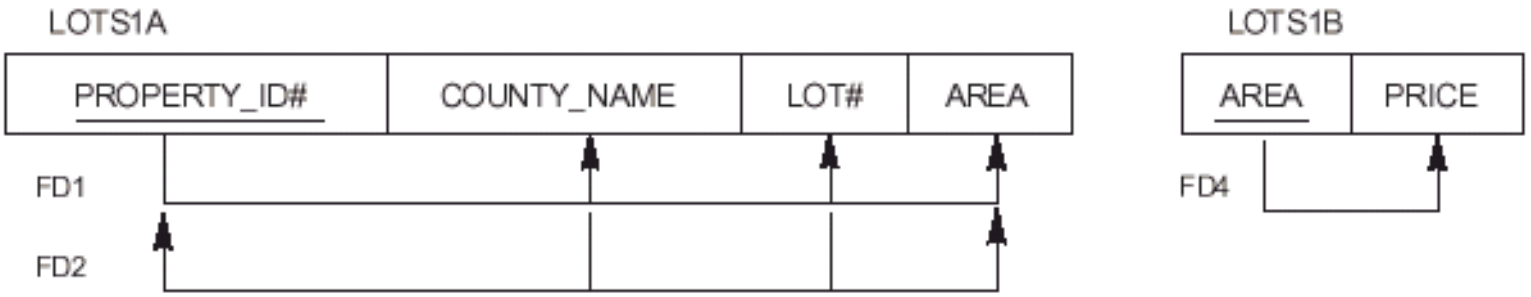
Forme Normali: *Esempio* (3)

- In accordo alla definizione generale di 3NF, LOTS2 è in 3NF, mentre FD4 in LOTS1 viola la 3NF.

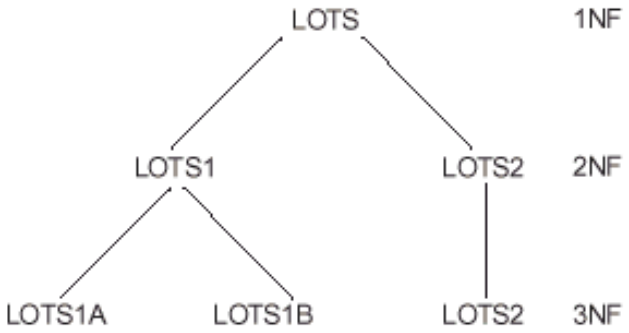
- Infatti AREA → PRICE, ma
  - AREA non è superchiave di LOTS1, e
  - PRICE non è attributo primo.

- Per normalizzare: decomponiamo LOTS1 rimuovendo PRICE e ponendolo insieme ad AREA in un altro schema di relazione.

Forme Normali: *Esempio* (4)



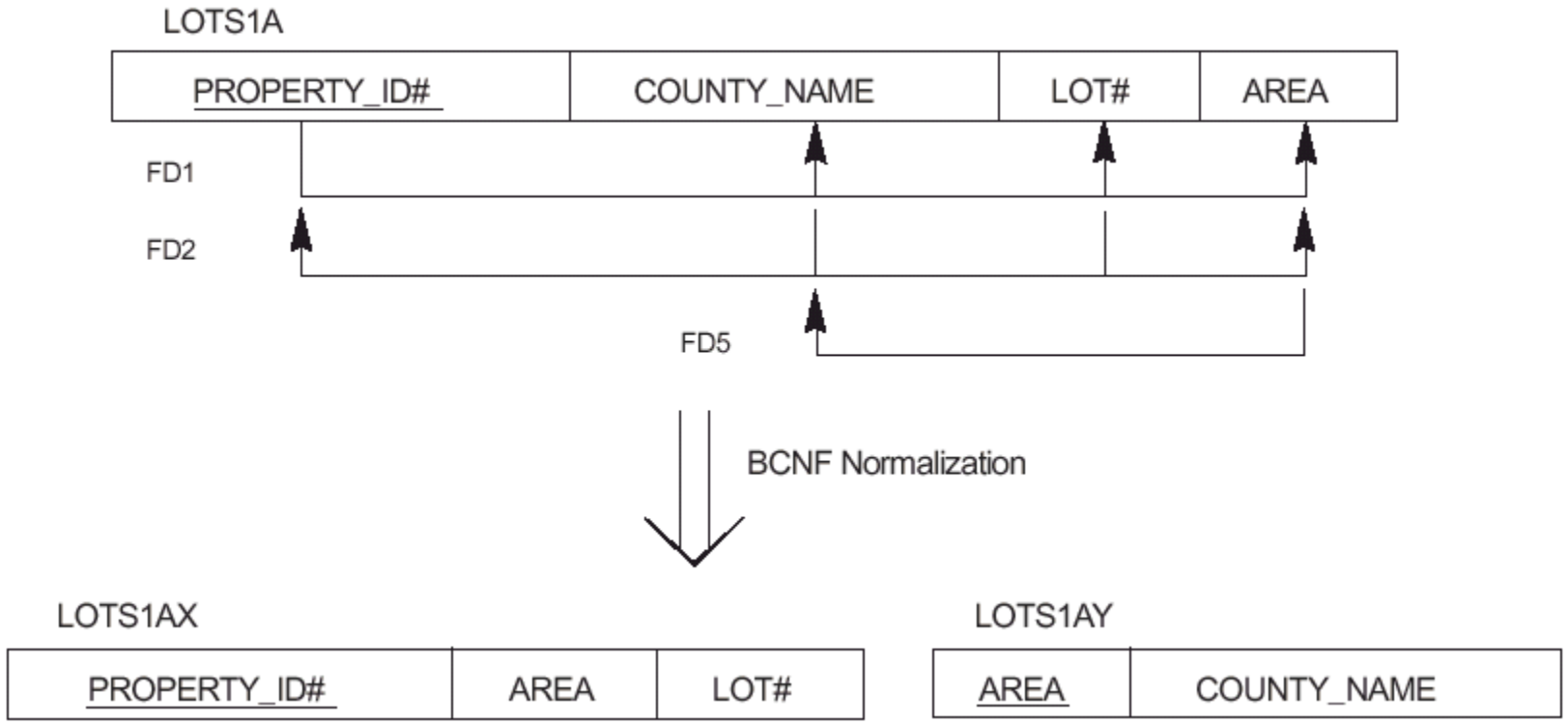
Decomposizione di LOTS1 in 3NF



Sommario delle decomposizioni

Forme Normali: *Esempio (5)*

- Supponiamo che ci siano migliaia di lotti: i lotti appartengono a solo due contee
  - Marion County con aree: 0,5 0,6 0,7 0,8 0,9 1,0
  - Liberty County con aree: 1,1 1,2 … 1,9 2,0



Forme Normali: *Esempio* (6)

TEACH

STUDENT	COURSE	INSTRUCTOR
Narayan	Database	Mark
Smith	Database	Navathe
Smith	Operating Systems	Ammar
Smith	Theory	Schulman
Wallace	Database	Mark
Wallace	Operating Systems	Ahamad
Wong	Database	Omiecinski
Zelaya	Database	Navathe

FD1: {STUDENT, COURSE} → INSTRUCTOR

FD2: INSTRUCTOR → COURSE

È in 3NF ma non in BCNF

Esistono varie decomposizioni

{STUDENT, INSTRUCTOR} e {STUDENT, COURSE}

{COURSE, INSTRUCTOR} e {COURSE , STUDENT}

{INSTRUCTOR, COURSE} e {INSTRUCTOR, STUDENT}

Tutte e tre perdono la dipendenza funzionale FD1