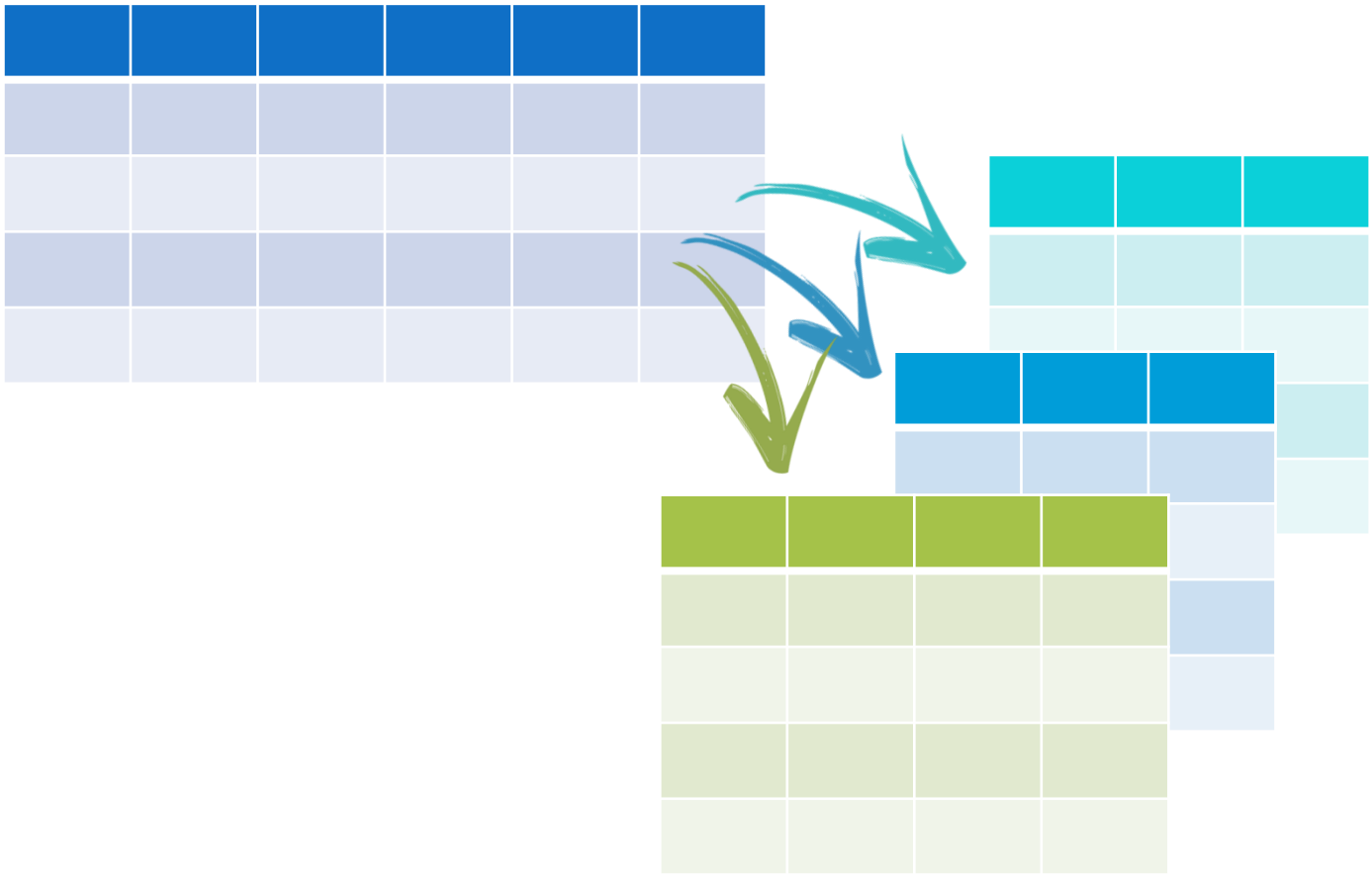


Normalizzazione

Annalisa Franco, Dario Maio

Università di Bologna

Forme normali

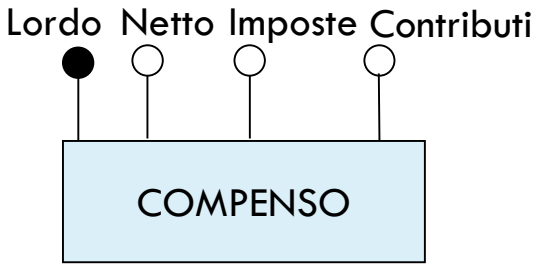
- Una forma normale è una proprietà di uno schema relazionale che ne garantisce la "qualità", cioè l'assenza di determinati difetti.
- Una relazione non normalizzata:
  - presenta ridondanze;
  - si presta a provocare anomalie di aggiornamento.
- Le forme normali sono di solito definite sul modello relazionale, ma rivestono un ruolo importante anche in altri contesti, ad esempio nel modello E/R.
- L'attività che permette di trasformare schemi non normalizzati in schemi che soddisfano una forma normale è detta normalizzazione.
- La normalizzazione deve essere utilizzata come tecnica di verifica dei risultati della progettazione di una base di dati.

- N.B. Alcuni argomenti ed esempi in queste slide sono stati derivati dal testo "Basi di dati" di Atzeni et al., McGraw-Hill.

Ridondanza concettuale

- Ridondanza concettuale: non vi sono replicazioni dello stesso dato, ma sono memorizzate informazioni che possono essere derivate da altre già contenute nel DB. Questi tipi di ridondanza si possono già presentare negli schemi E/R.

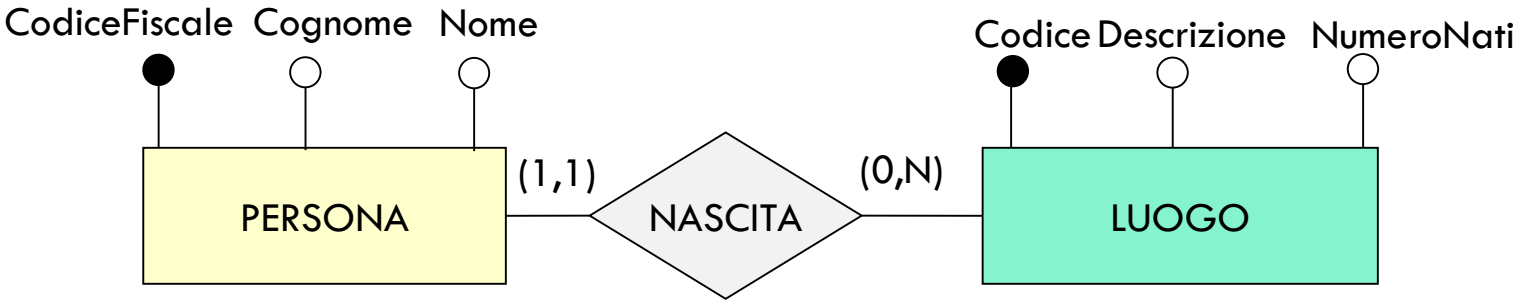
Esempio 1



L'importo netto del compenso può essere derivato come:
Netto = Lordo-Imposte-Contributi.

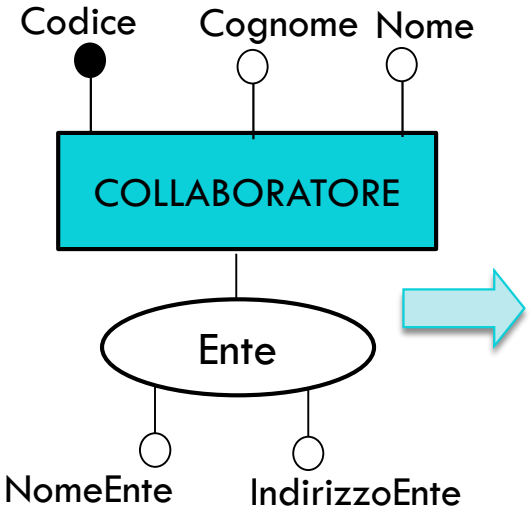
Esempio 2

Il numero dei nati per ogni città può essere calcolato contando le persone nate in quella città



Ridondanza logica

- Ridondanza logica: esistono duplicazioni sui dati che, oltre a comportare spreco di spazio di memoria, possono generare anomalie nelle operazioni sui dati.



Codice	Cognome	Nome	NomeEnte	IndirizzoEnte
29321	Bianchi	Giorgio	Fast	Via Corta 9, Bologna
24467	Rossi	Lucia	Spazio	Via Lunga 3, Cesena
26654	Verdi	Marco	Spazio	Via Lunga 3, Cesena
41132	Neri	Anna	Biomed	Via Larga 1, Bologna
11375	Viola	Carlo	Spazio	Via Lunga 3, Cesena

- Ridondanza: l'indirizzo di un ente è ripetuto in tutte le tuple dei suoi collaboratori.
- Anomalia di aggiornamento: se l'indirizzo di un ente cambia, è necessario modificare il valore in diverse tuple.
- Anomalia di inserimento: un nuovo ente senza collaboratori non può essere inserito.
- Anomalia di cancellazione: se si cancellano tutti i collaboratori afferenti a un ente si perdono le informazioni dell'ente stesso.

Ridondanza logica: una precisazione

- In un DB l'informazione può essere duplicata in modo :

NON RIDONDANTE:

la duplicazione dei dati **è necessaria,** l'eliminazione delle duplicazioni comporta perdita di informazione.

STUDENTI

Matricola	**Tutor**
125233	Maio
127988	Franco
150444	Franco
190787	Maio

duplicazione di dati non ridondante

RIDONDANTE:

la duplicazione dei dati **non è necessaria,** comporta spreco di memoria, è causa di possibili anomalie e inconsistenze.

STUDENTI

Matricola	**Tutor**	**Tel**
125233	Maio	7575
127988	Franco	5566
150444	Franco	5566
190787	Maio	7575

duplicazione di dati ridondante

Scomposizione di schemi

- Le ridondanze logiche si possono eliminare mediante scomposizione degli schemi. Nel seguito l'attenzione è concentrata su questo tipo di ridondanze. Useremo pertanto il termine ridondanza riferendoci alla ridondanza logica.

STUDENTI

Matricola	**Tutor**	**Tel**
125233	Maio	7575
127988	Franco	5566
150444	Franco	5566
190787	Maio	7575

STUDENTI_TUTOR

Matricola	**Tutor**
125233	Maio
127988	Franco
150444	Franco
190787	Maio

TUTOR

Tutor	**Tel**
Maio	7575
Franco	5566

Un altro esempio di relazione con anomalie

IMPIEGATI

Impiegato	Stipendio	Settore	Budget	Ruolo
Rossini	18000	Centro	2000000	tecnico
Verdoni	25000	Sud	1300000	venditore
Verdoni	25000	Nord	1500000	venditore
Bianconi	40000	Nord	1500000	direttore
Bianconi	40000	Sud	1300000	consulente
Bianconi	40000	Centro	2000000	consulente
Moretti	50000	Centro	2000000	direttore
Moretti	50000	Nord	1500000	venditore
Neri	47000	Nord	1500000	venditore
Neri	47000	Sud	1300000	direttore

- In un'unica relazione sono rappresentati gli impiegati con i relativi stipendi, i settori d'operatività con i relativi budget e il ruolo svolto dagli impiegati nei settori stessi.
- N.B. In generale le ridondanze logiche sono causate da errori durante la progettazione concettuale o da errate traduzioni di schemi E/R in schemi relazionali.

Analizziamo la relazione…

- Ogni impiegato ha un solo stipendio, anche se opera in più settori.
- Ogni settore ha un unico budget.
- Ogni impiegato in ciascun settore ricopre un solo ruolo, anche se può avere diversi ruoli in settori diversi.
- È stata utilizzata un'unica relazione per rappresentare tutte queste informazioni eterogenee:
  - gli impiegati con i relativi stipendi;
  - i settori con i relativi budget;
  - le partecipazioni degli impiegati ai settori con i relativi ruoli;
  - ciò comporta conseguenze non desiderabili nella gestione dei dati.

Ridondanze e anomalie

IMPIEGATI

Impiegato	Stipendio	Settore	Budget	Ruolo
~~Rossini~~	~~18000~~	~~Centro~~	~~2000000~~	~~tecnico~~
Verdoni	25000	Sud	1300000	venditore
Verdoni	25000	Nord	1500000	venditore
Bianconi	52000	Nord	1500000	direttore
Bianconi	52000	Sud	1300000	consulente
Bianconi	52000	Centro	2000000	consulente
Moretti	50000	Centro	2000000	direttore
Moretti	50000	Nord	1500000	venditore
Neri	47000	Nord	1500000	venditore
Neri	47000	Sud	1300000	direttore

??? ← Gialletti

- Lo stipendio di ciascun impiegato è ripetuto in tutte le tuple relative: **ridondanza**.
- Se lo stipendio di un impiegato varia, è necessario modificare il valore in diverse tuple: **anomalia di aggiornamento**.
- Se un impiegato interrompe la partecipazione a tutti i settori, dobbiamo cancellarlo: **anomalia di cancellazione**.
- Un nuovo impiegato senza attribuzione di un settore non può essere inserito: **anomalia di inserimento**.

Ridondanze e anomalie

- **Ridondanza**: presenza di dati ripetuti in diverse tuple senza aggiungere informazioni significative.
- **Anomalia di aggiornamento**: necessità di estendere l'aggiornamento di un dato a tutte le tuple in cui esso compare.
- **Anomalia di cancellazione**: l'eliminazione di una tupla, motivata dal fatto che non è più valido l'insieme dei concetti in essa espressi, può comportare l'eliminazione di dati che conservano comunque la loro validità.
- **Anomalia di inserimento**: l'inserimento di informazioni relative a uno solo dei concetti di pertinenza di una relazione è impossibile se non esiste un intero insieme di concetti in grado di costituire una tupla completa.

Dipendenza funzionale

- Per formalizzare i problemi visti si introduce un nuovo tipo di vincolo, la dipendenza funzionale (FD) tra attributi di una relazione.
- Si considerino:
  - uno schema di relazione R(T) e un'estensione r;
  - due sottoinsiemi (non vuoti) di T denominati X e Y rispettivamente.

- Si dice che in r vale la dipendenza funzionale $X \rightarrow Y$ (X determina funzionalmente Y) se

$$\forall\, t1,t2 \in r : t1[X] = t2[X] \Rightarrow t1[Y] = t2[Y]$$

cioè per ogni coppia di tuple t1 e t2 di r con gli stessi valori su X, t1 e t2 hanno gli stessi valori anche su Y.

Esempi di FD

IMPIEGATI	**Impiegato**	**Stipendio**	**Settore**	**Budget**	**Ruolo**

Nella relazione si hanno diverse FD, tra cui:

Impiegato → Stipendio

Settore → Budget

Impiegato, Settore → Ruolo

- Altre FD sono "meno interessanti" ("banali"), perché sono sempre soddisfatte, ad esempio:

Impiegato, Settore → Settore

  - Se $Y \subseteq X$ allora sicuramente $X \rightarrow Y$. FD di questo tipo sono dette FD banali.
  - $X \rightarrow Y$ è non banale se nessun attributo in Y appartiene a X.

FD - Precisazioni

- Le dipendenze funzionali rappresentano una generalizzazione dei vincoli di chiave.
- Una dipendenza funzionale è una caratteristica dello schema R(T), aspetto intensionale, e non della particolare estensione r dello schema, aspetto estensionale.
- Una dipendenza funzionale è dettata dalla semantica degli attributi di una relazione e non può essere inferita da una particolare estensione dello schema.
- Un'estensione di uno schema che rispetti una data dipendenza funzionale è detta estensione legale dello schema rispetto alla data dipendenza funzionale.
- Dire che $X \rightarrow Y$ significa asserire che i valori della componente Y dipendono (e dunque sono determinati) dai valori della componente X.
- Se $X \rightarrow Y$ non necessariamente risulta anche $Y \rightarrow X$.
- Se K è una chiave in uno schema R(T) allora ogni altro attributo di R(T) dipende funzionalmente da K. In altre parole K determina funzionalmente tutti gli attributi dello schema: posto T= KZ si ha $K \rightarrow Z$ e, poiché $K \rightarrow K$ si ha anche $K \rightarrow T$.

FD e Superchiavi

- Il concetto di superchiave si esprime facendo uso di FD.

$$K \subseteq T \text{ è superchiave di } R(T) \Leftrightarrow K \rightarrow T$$

Dimostrazione

- (se) Se $K \rightarrow T$ allora per ogni estensione legale r si ha che:

  $\forall\ t_1, t_2 \in r : t_1[K] = t_2[K] \Rightarrow t_1[T] = t_2[T]$, ovvero $t_1 = t_2$. Ciò equivale a dire che non possono esistere due tuple *distinte* con lo stesso valore di K.

- (solo se) Se K è superchiave di R(T), dalla definizione di superchiave si ha che:

  $t_1[K] = t_2[K] \Rightarrow t_1 = t_2$, e quindi $t_1[T] = t_2[T]$.

- N.B. Si ricorda che una chiave è una superchiave, ma non necessariamente una superchiave è anche chiave.

Anomalie e FD

- Le anomalie viste si riconducono alla presenza delle FD:

  Impiegato → Stipendio

  Settore → Budget

- viceversa non causa problemi la FD:

  Impiegato, Settore → Ruolo

- Motivazioni:
  - la terza FD ha sulla sinistra una chiave e non causa anomalie;
  - le prime due FD non hanno sulla sinistra una chiave e causano anomalie.

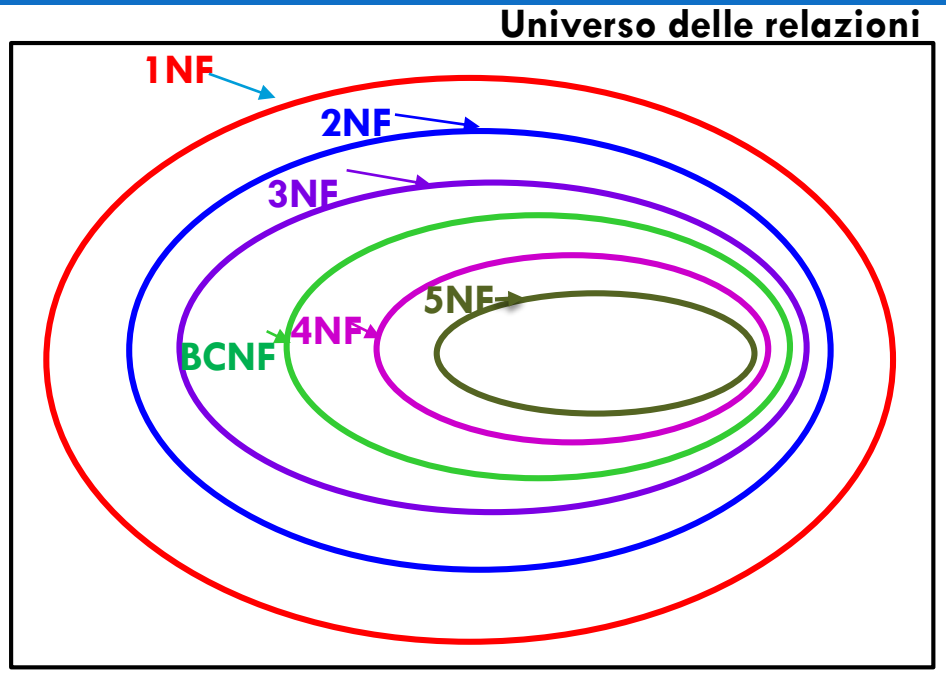
- La relazione contiene alcune informazioni legate alla chiave e altre ad attributi che non formano una chiave.

Forme normali

Si definiscono **UNF** (**U**n - **N**ormalized **F**orm) le relazioni che non sono conformi a nessuna forma normale.

Classica gerarchia delle forme normali

1NF (First Normal Form)
2NF (Second Normal Form)
3NF (Third Normal Form)
BCNF (Boyce–Codd Normal Form)
4NF (Fourth Normal Form)
5NF (Fifth Normal Form)



- Il processo di normalizzazione fu inizialmente introdotto da Codd: nel 1970 con la definizione del modello relazionale, nel 1971 con la definizione di 1NF e nel 1972 con la definizione di 2NF e 3NF. Nel 1974 Boyce e Codd definirono una forma più restrittiva di 3NF denominata BCNF. Tutte queste forme normali si basano sulle dipendenze funzionali tra gli attributi di una relazione.
- Più tardi a cura di Fagin furono definite le forme normali 4NF (1977) e 5NF (1979) basate rispettivamente sulle dipendenze multivalore e sulle dipendenze di join, e successivamente ulteriori forme normali.

1NF: definizione

First Normal Form (1NF)

Uno schema R(T) è in 1NF se e solo se ***il dominio di ciascun attributo comprende solo valori atomici*** (semplici, indivisibili) e il valore di ciascun attributo in una tupla è un valore singolo del dominio di quell'attributo.

- Dunque 1NF non permette "relazioni dentro relazioni" e "relazioni come attributi di tuple". I soli valori di attributi ammissibili sono i singoli valori atomici (non ulteriormente decomponibili a parte funzioni speciali) rispetto al RDMBS (definizione di atomicità secondo Codd).
- Oggi 1NF è considerata parte integrante della definizione formale di relazione del modello relazionale di base.
- Con non-scomponibilità di un attributo non dobbiamo intendere che il valore dell'attributo non possa essere suddiviso in sotto-parti (ad esempio, l'indirizzo può essere scomposto in "Via", "Sacchi", "3", o addirittura in tutti i caratteri che lo compongono). Quello che importa è che ogni valore dell'attributo sia dal punto di vista semantico un'informazione unica: ad esempio, non si possono inserire due sedi per l'attributo "Sede" in quanto si tratta di due informazioni semanticamente distinte. In pratica, riprendendo il concetto di insieme, un attributo può assumere uno e un solo valore, preso fra gli elementi del suo dominio.

1NF: esempio A (1)

- Si consideri lo schema DIPARTIMENTI(CodDip, Nome, CodDir, SediDip) e lo stato:

CodDip	**Nome**	**CodDir**	**SediDip**
D0001	Amministrazione	33301	(Milano, Napoli, Roma)
D0005	Produzione	18007	Aprilia
D0003	Ricerca	33010	Napoli

- La relazione non è in 1NF a causa dell'attributo SediDip. Sono possibili due interpretazioni:
  1. il dominio di SediDip contiene valori atomici ma alcune tuple hanno un insieme di questi valori, in questo caso SediDip non dipende funzionalmente da CodDip;
  2. Il dominio di SediDip contiene insiemi di valori e perciò non è atomico; in questo caso CodDip $\rightarrow$ SediDip poiché ogni insieme è considerato un unico membro del dominio dell'attributo, ovvero il dominio di SediDip è l'insieme potenza dell'insieme delle singole sedi.

1NF: esempio A (2)

Soluzione 1: si espande la chiave in modo da avere tuple separate per ogni sede differente di un dipartimento.

DIPARTIMENTI(CodDip, Nome, CodDir, SedeDip)

CodDip	**Nome**	**CodDir**	**SedeDip**
D0001	Amministrazione	33301	Milano
D0001	Amministrazione	33301	Napoli
D0001	Amministrazione	33301	Roma
D0005	Produzione	18007	Aprilia
D0003	Ricerca	33010	Napoli

Questa soluzione ha lo svantaggio di inserire ridondanza d'informazione.

1NF: esempio A (3)

Soluzione 2: se è noto a priori il numero massimo N di sedi che può avere un dipartimento si può sostituire l'attributo SediDip con N attributi separati, ad esempio nel caso di N=3:

DIPARTIMENTI(CodDip, Nome, CodDir, Sede1, Sede2, Sede3)

CodDip	**Nome**	**CodDir**	**Sede1**	**Sede2**	**Sede3**
D0001	Amministrazione	33301	Milano	Napoli	Roma
D0005	Produzione	18007	Aprilia	NULL	NULL
D0003	Ricerca	33010	Napoli	NULL	NULL

Questa soluzione ha lo svantaggio di introdurre valori nulli.

1NF: esempio A (4)

Soluzione 3: si rimuove l'attributo SediDip e lo si pone in un'altra relazione separata con chiave combinazione di CodDip e SedeDip.

DIPARTIMENTI(CodDip, Nome, CodDir)

CodDip	Nome	CodDir
D0001	Amministrazione	33301
D0005	Produzione	18007
D0003	Ricerca	33010

SEDI(CodDip:DIPARTIMENTI, SedeDip)

CodDip	SedeDip
D0001	Milano
D0001	Napoli
D0001	Roma
D0005	Aprilia
D0003	Napoli

Questa soluzione non presenta ridondanze ed è completamente generale, non presentando limiti sul massimo numero di sedi per un dipartimento.

N.B. Nel seguito, laddove non vi sia ambiguità, per semplicità e per motivi di spazio nelle slide si omette l'indicazione estesa delle foreign key; esempio:

SEDI(CodDip, SedeDip) invece di SEDI(CodDip:DIPARTIMENTI, SedeDip)

1NF: esempio B (1)

Se fosse concesso di avere relazioni nidificate si potrebbe definire lo schema CARTELLINI_ORE_LAVORATE:

CodImpiegato	**Cognome**	**Nome**	**CodProgetto**	**Data**	**OreLavorate**

Una tupla rappresenta un impiegato e una relazione (che riepiloga le ore lavorate da quell'impiegato nei vari progetti in varie date).

Possibile estensione

0012	Rossi	Giorgia	1023	05/07/2010	5
			1225	15/07/2010	6
			1225	15/10/2010	8
0115	Bianchi	Mario	1023	08/07/2010	7
			1128	17/09/2010	3
0085	Verdi	Luigi	1023	05/07/2010	4
			1023	06/07/2010	6

{CodProgetto, Data} è la chiave primaria parziale della relazione nidificata, ovvero {CodProgetto, Data} deve esibire valori unici all'interno di ogni tupla della relazione nidificata.

1NF: esempio B (2)

La normalizzazione in 1NF porta a progettare gli schemi:

IMPIEGATI

CodImpiegato	**Cognome**	**Nome**

CARTELLINI

CodImpiegato	**CodProgetto**	**Data**	**OreLavorate**

A livello di estensioni:

0012	Rossi	Giorgia
0015	Bianchi	Mario
0085	Verdi	Luigi

0012	1023	05/07/2010	5
0012	1225	15/07/2010	6
0012	1225	15/10/2010	8
0015	1023	08/07/2010	7
0015	1128	17/09/2010	3
0085	1023	05/07/2010	4
0085	1023	06/07/2010	6

Si spostano gli attributi della relazione nidificata in una nuova relazione e si propaga la chiave primaria della relazione originaria.

La nuova relazione ha come chiave primaria la combinazione della chiave parziale e della chiave primaria della relazione originaria.

1NF ma non 2NF: un esempio

- Si consideri lo schema:

  MAGAZZINI(Articolo, Magazzino, Quantità, Indirizzo)

- i vincoli (FD):

  Articolo, Magazzino → Quantità, Indirizzo (AM → QI)

  Magazzino → Indirizzo (M → I)

- e lo stato legale:

Articolo	**Magazzino**	**Quantità**	**Indirizzo**
scarpe	VR1	25000	via Albere 17 - Verona
pantaloni	VR1	18000	via Albere 17 - Verona
scarpe	BO1	4500	via Agucchi 3 - Bologna
camicie	VR2	7000	via Monti 6 - Verona

I problemi sono dovuti a M → I:
ogni tupla memorizza informazioni individuate da un valore della chiave AM, ma l'indirizzo dipende solo parzialmente dalla chiave.

2NF: definizione

- Attributo primo: dato uno schema R(T), un attributo A∈T è primo se e solo se fa parte di almeno una chiave dello schema. In caso contrario è detto non-primo.
- Nello schema

  MAGAZZINI(Articolo, Magazzino, Quantità, Indirizzo)

  Articolo e Magazzino sono primi, Quantità e Indirizzo sono non-primi.

Second Normal Form (2NF)

Uno schema R(T) con vincoli F è in 2NF se e solo se ***ogni attributo non-primo dipende completamente (non parzialmente) da ogni chiave candidata dello schema***, ovvero se ***non c'è dipendenza parziale di un attributo non-primo da una chiave***.

- Uno schema in 1NF le cui chiavi siano tutte "semplici", ovvero formate da un singolo attributo, è anche in 2NF.

Normalizzazione in 2NF

- La soluzione consiste nell'estrarre la FD che crea i problemi, generando gli schemi:

  ARTICOLI_IN_MAGAZZINI(Articolo, Magazzino, Quantità) (AM → Q)

  INDIRIZZI_MAGAZZINI(Magazzino, Indirizzo) (M → I)

Articolo	Magazzino	Quantità
scarpe	VR1	25000
pantaloni	VR1	18000
scarpe	BO1	4500
camicie	VR2	7000

Magazzino	Indirizzo
VR1	via Albere 17 - Verona
BO1	via Agucchi 3 - Bologna
VR2	Via Monti 6 - Verona

L'informazione originale si può ricostruire eseguendo un join tra le due relazioni:

MAGAZZINI = ARTICOLI_IN_MAGAZZINI ▷◁ INDIRIZZI_MAGAZZINI

2NF e chiavi candidate

Una relazione in cui non vi sono dipendenze funzionali parziali dalla chiave primaria è tipicamente in 2NF ma non sempre. Si consideri ad esempio lo schema PRODUTTORI (Produttore, Modello, NomeModelloCompleto, Stato) e una sua estensione legale.

Produttore	**Modello**	**NomeModelloCompleto**	**Stato**
Forte	X-Prime	F X-Prime	Italia
Forte	Ultraclean	F Ultraclean	Italia
Dent-o-Fresh	EZbrush	DoF EZBrush	USA
Kobayashi	ST-60	K ST-60	Giappone
Hoch	Toothmaster	H Toothmaster	Germania
Hoch	X-Prime	H X-Prime	Germania

- La relazione non è in 2NF anche se il progettista ha scelto come chiave primaria {NomeModelloCompleto}. Una chiave candidata è anche {Produttore, Modello} ma Produttore → Stato (dipendenza parziale).
- La trasformazione in 2NF prevede due relazioni:

PRODUTTORI_SPAZZOLINI(Produttore, Stato)

MODELLI_SPAZZOLINI(Produttore, Modello, NomeModelloCompleto)

Ancora anomalie

Con riferimento agli impiegati di una banca con diverse agenzie, si consideri il seguente schema in 2NF:

IMPIEGATI(IdImpiegato, Cognome, Nome, Agenzia, Luogo)

- con vincoli (FD):

  IdImpiegato → Cognome, Nome, Agenzia, Luogo (I → CNAL)

  Agenzia → Luogo (A → L)

- e l'estensione legale:

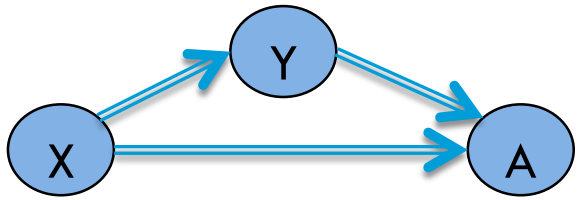
IdImpiegato	Cognome	Nome	Agenzia	Luogo
001	Rossi	Carlo	02400	Bologna
002	Verdi	Maria	53880	Bergamo
003	Bianchi	Giulia	04826	Cagliari
004	Neri	Franco	23900	Cesena
005	Gialli	Marco	02400	Bologna

I problemi sono dovuti a A → L: infatti L dipende transitivamente dalla chiave I.

3NF: definizione

Dipendenza transitiva: dato uno schema R(T), $X \subseteq T$, $A \in T$, A dipende transitivamente da X se esiste $Y \subset T$ tale che:

1. $X \rightarrow Y$ {X determina Y}
2. $\neg (Y \rightarrow X)$ {Y non determina X}
3. $Y \rightarrow A$ {Y determina A....}
4. $A \notin Y$ {...non banalmente}



Third Normal Form (3NF)

Uno schema R(T) con vincoli F è in 3NF se e solo se ***ogni attributo non-primo non dipende transitivamente da nessuna chiave*** ovvero se ***non c'è dipendenza transitiva di un attributo non-primo da una chiave***.

N.B. Nel seguito, per semplicità, chiameremo impropriamente dipendenza transitiva una FD del tipo $Y \rightarrow A$, quando a causa di $X \rightarrow Y$ si genera una catena $X \rightarrow Y \rightarrow A$, per cui A viene a dipendere transitivamente da X. In realtà, come da definizione, la vera dipendenza transitiva è $X \rightarrow A$.

Si deve comunque intendere con questa notazione che $Y \rightarrow A$ genera una dipendenza transitiva di A da X, perché $X \rightarrow Y$.

Esempio di normalizzazione in 3NF

- Con riferimento allo schema IMPIEGATI(IdImpiegato,Cognome, Nome, Agenzia, Luogo) la soluzione consiste nell'estrarre la FD che crea i problemi, generando gli schemi:

  IMPIEGATI_AGENZIE(IdImpiegato,Cognome,Nome,Agenzia) (I → CNA)

  AGENZIE(Agenzia,Luogo) (A → L)

IdImpiegato	Cognome	Nome	Agenzia
001	Rossi	Carlo	02400
002	Verdi	Maria	53880
003	Bianchi	Giulia	04826
004	Neri	Franco	23900
005	Gialli	Marco	02400

Agenzia	Luogo
02400	Bologna
53880	Bergamo
04826	Cagliari
23900	Cesena

- L'informazione originale si può ricostruire eseguendo un join naturale tra le due relazioni:

  IMPIEGATI = IMPIEGATI_AGENZIE ▷◁ AGENZIE

Un altro esempio di normalizzazione in 3NF

IMPIEGATI

Impiegato	**Stipendio**	**Settore**	**Budget**	**Ruolo**

Lo schema non è normalizzato (non è in 3NF né in 2NF), la soluzione consiste nel "decomporlo" sulla base delle FD.

Impiegato → Stipendio

Impiegato	**Stipendio**
Rossini	18000
Verdoni	25000
Bianconi	40000
Moretti	50000
Neri	47000

Settore → Budget

Settore	**Budget**
Nord	1500000
Centro	2000000
Sud	1300000

Impiegato, Settore → Ruolo

Impiegato	**Settore**	**Ruolo**
Rossini	Centro	tecnico
Verdoni	Sud	venditore
Verdoni	Nord	venditore
Bianconi	Nord	direttore
Bianconi	Sud	consulente
Bianconi	Centro	consulente
Moretti	Centro	direttore
Moretti	Nord	venditore
Neri	Nord	venditore
Neri	Sud	direttore

3NF: definizione equivalente

- La definizione di 3NF data in precedenza si basa sulle dipendenze transitive di attributi non-primi dalle chiavi.
- Una definizione equivalente che non utilizza FD transitive, spesso riportata in molti testi, è riportata di seguito.

Terza Forma Normale

Uno schema R(T) con vincoli F è in 3NF se e solo se, per ogni dipendenza funzionale non banale $X \rightarrow Y$ definita su R(T), X è una superchiave di R(T) oppure ogni attributo A in Y è contenuto in almeno una chiave di R(T), cioè A è un attributo primo.

Esempio IMPIEGATI(IdImpiegato, Cognome, Nome, Agenzia, Luogo)

- Lo schema non è in 3NF in quanto nella dipendenza Agenzia $\rightarrow$ Luogo si ha che Agenzia non è superchiave e Luogo è un attributo non-primo.

Esercizio riepilogativo

Si consideri il seguente schema relazionale:

TEST_LAB (MatrStudente, NomeStudente, CodCorso, NomeCorso, CodTitolare, NomeTitolare, CodEsaminatore, NomeEsaminatore, DataProva, Voto)

Sono registrate anche eventuali prove non superate. Un corso ha un solo professore titolare che non coincide necessariamente con il professore esaminatore. Si evidenzino tutte le dipendenze funzionali non banali e le problematiche presenti nello schema. Qualora lo schema non sia in 3NF, si determini un insieme di schemi che siano in 3NF e risultino equivalenti, dal punto di vista informativo, allo schema dato.

Soluzione: individuare le FD

FD_1 MatrStudente → NomeStudente

FD_2 CodCorso → NomeCorso

FD_3 CodCorso → CodTitolare

Dipendenze parziali

FD_4 CodTitolare → NomeTitolare

FD_5 CodEsaminatore → NomeEsaminatore

Dipendenze transitive

A causa della presenza di dipendenze funzionali parziali la relazione non è in 2NF. Per ottenere una relazione in 2NF si devono spezzare le dipendenze parziali. Per ottenere una relazione in 3NF si devono poi risolvere le dipendenze funzionali transitive.

Esercizio riepilogativo: 2NF

TEST_LAB (MatrStudente, NomeStudente, CodCorso, NomeCorso, CodTitolare, NomeTitolare, CodEsaminatore, NomeEsaminatore, DataProva, Voto)

FD_1 MatrStudente → NomeStudente
FD_2 CodCorso → NomeCorso
FD_3 CodCorso → CodTitolare

Dipendenze parziali

Soluzione: normalizzare in 2NF

Spezzando le dipendenze parziali si ottengono gli schemi in 2NF:

PROVE_LAB (MatrStudente: STUDENTI, CodCorso: CORSI, CodEsaminatore, NomeEsaminatore, DataProva, Voto)

STUDENTI (MatrStudente, NomeStudente)

CORSI (CodCorso, NomeCorso, CodTitolare, NomeTitolare)

Esercizio riepilogativo: 3NF

PROVE_LAB (MatrStudente, CodCorso, CodEsaminatore, NomeEsaminatore, DataProva,Voto)

STUDENTI (MatrStudente, NomeStudente)

CORSI (CodCorso, NomeCorso, CodTitolare, NomeTitolare)

FD_4 CodTitolare → NomeTitolare
FD_5 CodEsaminatore → NomeEsaminatore

} Dipendenze transitive

Soluzione: normalizzare in 3NF

Spezzando le dipendenze transitive si ottengono gli schemi in 3NF:

PROVE_LAB (MatrStudente:STUDENTI,CodCorso:CORSI,CodEsaminatore:PROFESSORI, DataProva, Voto)

PROFESSORI (CodProfessore, NomeProfessore)

(N.B. Include Esaminatori e Titolari)

STUDENTI (MatrStudente, NomeStudente)

CORSI (CodCorso, NomeCorso, CodTitolare: PROFESSORI)

Esempio di decomposizione con perdita

Venditore	Agenzia	Sede
Rossetti	Spazio	Roma
Verdoni	Fast	Milano
Verdoni	Service	Milano
Moretti	Centrale	Milano
Moretti	Service	Milano

Venditore	Sede
Rossetti	Roma
Verdoni	Milano
Moretti	Milano

Agenzia	Sede
Centrale	Milano
Fast	Milano
Service	Milano
Spazio	Roma

con FD: Venditore → Sede

Agenzia → Sede

Se si esegue il join naturale dei due schemi ottenuti decomponendo come sopra, la relazione ricostruita è diversa da quella di partenza.

tuple spurie

Venditore	Agenzia	Sede
Rossetti	Spazio	Roma
Verdoni	Centrale	Milano
Verdoni	Fast	Milano
Verdoni	Service	Milano
Moretti	Centrale	Milano
Moretti	Fast	Milano
Moretti	Service	Milano

Decomposizione senza perdita

- La decomposizione non deve assolutamente alterare il contenuto informativo del DB.
- Si introduce pertanto il seguente requisito:

Decomposizione senza perdita (lossless)

Uno schema R(X) si decompone senza perdita negli schemi $R_1(X_1)$ e $R_2(X_2)$ se, ***per ogni stato legale r su R(X), il join naturale delle proiezioni di r su X_1 e X_2 è uguale a r stessa***:

$$\pi_{X_1}(r) \bowtie \pi_{X_2}(r) = r$$

- Una decomposizione con perdita può generare tuple spurie.
- Per decomporre senza perdita è necessario e sufficiente che il join naturale sia eseguito su una superchiave di uno dei due sottoschemi, ovvero che valga:

$$X1 \cap X2 \rightarrow X1 \text{ oppure } X1 \cap X2 \rightarrow X2$$

Esempio di decomposizione lossless

Una decomposizione che non altera il contenuto informativo è:

Venditore	Agenzia	Sede
Rossetti	Spazio	Roma
Verdoni	Fast	Milano
Verdoni	Service	Milano
Moretti	Centrale	Milano
Moretti	Service	Milano

Venditore	Sede
Rossetti	Roma
Verdoni	Milano
Moretti	Milano

Venditore	Agenzia
Rossetti	Spazio
Verdoni	Fast
Verdoni	Service
Moretti	Centrale
Moretti	Service

OK!

... ma i problemi non sono ancora finiti…

Modifica con violazione di una FD

... supponiamo di voler effettuare una modifica:

Moretti assegnato anche all'agenzia Spazio.

Venditore	Sede
Rossetti	Roma
Verdoni	Milano
Moretti	Milano

Venditore	Agenzia
Rossetti	Spazio
Verdoni	Fast
Verdoni	Service
Moretti	Centrale
Moretti	Service
Moretti	Spazio

... ricostruendo la relazione otteniamo:

Venditore	Agenzia	Sede
Rossetti	Spazio	Roma
Verdoni	Fast	Milano
Verdoni	Service	Milano
Moretti	Centrale	Milano
Moretti	Service	Milano
Moretti	Spazio	Milano

che viola la FD Agenzia → Sede

Ancora anomalie

- Si consideri lo schema ELENCO_TEL(Pref, Num, Località, Abbonato, TopCiv) con vincoli:
  - Pref, Num → Località, Abbonato, TopCiv (PN → LAT)
  - Località → Pref (L → P)

  chiavi candidate {Pref, Numero} e {Località, Numero}
- Nella seguente estensione legale l'informazione sul prefisso è replicata per ogni abbonato:

ELENCO_TEL

Pref	**Numero**	**Località**	**Abbonato**	**TopCiv**
051	432175	Bologna	Rossi M.	Via Mazzini 124
059	272225	Modena	Bianchi G.	Via Emilia 233
051	227951	Bologna	Rossi M.	Via Amendola 14
051	314255	Castenaso	Neri E.	Via Mazzini 7
059	227951	Vignola	Verdi P.	Piazza Roma 14

Lo schema è in 3NF, in quanto Pref è primo (non vi è una dipendenza transitiva).

BCNF

- 3NF mira a risolvere i problemi causati da dipendenze transitive per attributi non-primi.
- BCNF è una forma normale più restrittiva di 3NF; essa estende le considerazioni sinora svolte anche agli attributi primi.

Forma Normale di Boyce-Codd (BCNF)

Uno schema R(T) con vincoli F è in BCNF se, ***per ogni dipendenza funzionale (non banale) $X \rightarrow Y$ definita su di esso, X è una superchiave di R(T)***.

- Lo schema ELENCO_TELEFONICO(Pref, Num, Località, Abbonato, TopCiv) con vincoli:
  - Pref,Num $\rightarrow$ Località, Abbonato, TopCiv (PN $\rightarrow$ LAT)
  - Località $\rightarrow$ Pref (L $\rightarrow$ P)

  non è in BCFN a causa della FD Località $\rightarrow$ Pref, infatti Pref è un attributo primo ma Località non è superchiave.

Una decomposizione non corretta

- La seguente decomposizione non è corretta, poiché non è lossless:

  NUM_TEL(Pref, Num,Abbonato,TopCiv)

  PREF_TEL(Località,Pref)

  L'attributo importato in Pref non è la chiave della relazione PREF_TEL.

- Non è possibile risalire univocamente all'indirizzo dell'abbonato (in presenza di più località con lo stesso prefisso).

NUM_TEL

Pref	Numero	Abbonato	TopCiv
051	432175	Rossi M.	Via Mazzini 124
059	272225	Bianchi G.	Via Emilia 233
051	227951	Rossi M.	Via Amendola 14
051	314255	Neri E.	Via Mazzini 7
059	227951	Verdi P.	Piazza Roma 14

PREF_TEL

Pref	Località
051	Bologna
059	Modena
051	Castenaso
059	Vignola

Dove vive l'abbonato "Rossi M."? A Bologna o a Castenaso?

Una decomposizione corretta

- Una soluzione corretta consiste nel decomporre lo schema in:

  NUM_TEL(Num,Località,Abbonato,TopCiv)

  PREF_TEL(Località,Pref)

Numero	Località	Abbonato	TopCiv
432175	Bologna	Rossi M.	Via Mazzini 124
272225	Modena	Bianchi G.	Via Emilia 233
227951	Bologna	Rossi M.	Via Amendola 14
314255	Castenaso	Neri E.	Via Mazzini 7
227951	Vignola	Verdi P.	Piazza Roma 14

Pref	Località
051	Bologna
059	Modena
051	Castenaso
059	Vignola

La decomposizione è lossless infatti:

$$(\text{NUM\_TEL} \bowtie \text{PREF\_TEL}) = \text{ELENCO\_TELEFONICO}$$

ma presenta ancora problemi…

…modifichiamo il DB…

Numero	Località	Abbonato	TopCiv
432175	Bologna	Rossi M.	Via Mazzini 124
272225	Modena	Bianchi G.	Via Emilia 233
227951	Bologna	Rossi M.	Via Amendola 14
314255	Castenaso	Neri E.	Via Mazzini 7
227951	Vignola	Verdi P.	Piazza Roma 14
227951	Modena	Gialli E.	Via Milano 4

Pref	Località
059	Modena
051	Bologna
051	Castenaso
059	Vignola

Supponiamo di voler inserire un nuovo abbonato Gialli:

Se si ricostruisce la relazione originaria si ottengono 2 tuple con lo stesso numero di telefono:

Pref	Numero	Località	Abbonato	TopCiv
051	432175	Bologna	Rossi M.	Via Mazzini 124
059	272225	Modena	Bianchi G.	Via Emilia 233
051	227951	Bologna	Rossi M.	Via Amendola 14
051	314255	Castenaso	Neri E.	Via Mazzini 7
059	227951	Vignola	Verdi P.	Piazza Roma 14
059	227951	Modena	Gialli E.	Via Milano 4

Attenzione ai vincoli!

- Una estensione legale nello schema decomposto genera sullo schema ricostruito (NUM_TEL ▷◁ PREF_TEL) una soluzione non ammissibile.

- Ogni singola estensione è "localmente" legale, ma il DB "globalmente" non lo è, infatti esistono in questo caso due abbonati (Verdi P. e Gialli E.) che hanno lo stesso numero di telefono (059-227951).

- Problemi di consistenza dei dati si hanno quando la decomposizione "separa" gli attributi di una FD. Per verificare che la FD sia rispettata si rende necessario far riferimento a entrambe le relazioni.

- La FD Pref, Num → Località non è rispettata nel DB e nessuno dei due schemi include tutti e tre gli attributi.

Preservazione delle dipendenze

- Si dice che una decomposizione **preserva le dipendenze** se ciascuna delle dipendenze funzionali dello schema originario coinvolge attributi che compaiono tutti insieme in uno degli schemi decomposti:
  - nell'esempio Pref, Num → Località non è conservata.
- Se una FD non si preserva diventa più complicato capire quali sono le modifiche del DB che non violano la FD stessa.
- In generale, prima di effettuare una modifica, si devono eseguire query SQL di verifica.

Esempio di query di verifica (A)

- Bisogna verificare che la FD Pref, Num → Località sia conservata, a tal fine per inserire un nuovo abbonato occorre controllare che non esista nessun altro abbonato in una località con lo stesso prefisso di Modena che abbia lo stesso numero di telefono 227951.

Numero	**Località**	…
432175	Bologna	
272225	Modena	
227951	Bologna	
314255	Castenaso	
227951	Vignola	

N

Pref	**Località**
059	Modena
051	Bologna
051	Castenaso
059	Vignola

P1

Pref	**Località**
059	Modena
051	Bologna
051	Castenaso
059	Vignola

P2

```
SELECT *  -- OK se non restituisce alcuna tupla
FROM    NUM_TEL N
WHERE   N.Numero = '227951'
AND     N.Località IN ( SELECT  P2.Località
                        FROM    PREF_TEL P1, PREF_TEL P2
                        WHERE   P1.Pref = P2.Pref
                        AND     P1.Località = 'Modena')
```

Esempio di query di verifica (B)

- Con riferimento all'esempio precedente per evitare che l'inserimento del fatto «Moretti assegnato all'agenzia Spazio» provochi la violazione della FD Agenzia → Sede, si deve verificare che l'agenzia (Spazio) sia presso la stessa sede del venditore (Moretti). A tal fine si deve trovare un venditore che lavora nell'agenzia Spazio.

VENDITORI

Venditore	Sede
Rossetti	Roma
Verdoni	Milano
Moretti	Milano

VENDITORI_AGENZIE

Venditore	Agenzia
Rossetti	Spazio
Verdoni	Fast
Verdoni	Service
Moretti	Centrale
Moretti	Service

```
SELECT *  -- OK se restituisce una tupla
FROM      VENDITORI V
WHERE     V.Venditore = 'Moretti'
  AND     V.Sede IN ( SELECT V1.Sede
                        FROM   VENDITORI V1, VENDITORI_AGENZIE VA
                        WHERE  V1.Venditore = VA.Venditore
                        AND  VA.Agenzia = `Spazio')
```

Qualità di una decomposizione

- Benché gli schemi in 3NF non siano esenti da problemi, questo livello di normalizzazione è comunemente accettato nella pratica.
- Nel caso generale, problemi di complessità computazionale rendono improponibile affrontare l'attività di normalizzazione mediante tecniche di "analisi". I seguenti problemi sono NP-completi:
  - determinare se un attributo è primo;
  - verificare se esiste una chiave di grado minore di k (k costante);
  - verificare se uno schema è in 3NF rispetto a un insieme di FD.
- L'approccio adottato è di tipo costruttivo, ovvero anziché verificare se uno schema è al livello di normalizzazione desiderato, si progettano schemi conformi a tale livello.
- Qualità di una decomposizione (ottenibile con algoritmi di normalizzazione):
  - **deve essere senza perdita**, per garantire la ricostruzione delle informazioni originarie;
  - **dovrebbe preservare le dipendenze**, per semplificare il mantenimento dei vincoli di integrità originari.

Qualità di una decomposizione: esempio

Con riferimento all'esempio precedente, ciò suggerisce di inserire un'ulteriore relazione AGENZIE che consente di verificare più facilmente il rispetto della dipendenza funzionale Agenzia → Sede.

VENDITORI_AGENZIE_SEDI

Venditore	Agenzia	Sede
Rossetti	Spazio	Roma
Verdoni	Fast	Milano
Verdoni	Service	Milano
Moretti	Centrale	Milano
Moretti	Service	Milano

VENDITORI

Venditore	Sede
Rossetti	Roma
Verdoni	Milano
Moretti	Milano

VENDITORI_AGENZIE

Venditore	Agenzia
Rossetti	Spazio
Verdoni	Fast
Verdoni	Service
Moretti	Centrale
Moretti	Service

AGENZIE

Agenzia	Sede
Spazio	Roma
Fast	Milano
Service	Milano
Centrale	Milano

La query di verifica è ora più semplice

```
SELECT *  -- OK se restituisce una tupla
FROM      VENDITORI V, AGENZIE A
WHERE     V.Venditore = 'Moretti'
  AND     A.Agenzia = `Spazio'
  AND     V.Sede = A.Sede
```

Algoritmo di decomposizione in 3NF

- L'idea alla base dell'algoritmo di sintesi che produce una decomposizione in 3NF consiste nel creare una relazione per ogni gruppo di FD che hanno lo stesso lato sinistro (determinante) e inserire nello schema corrispondente gli attributi coinvolti in almeno una FD del gruppo.

  Esempio: se le FD individuate sullo schema R(ABCDEFG) sono:

  $$AB \rightarrow CD,\ AB \rightarrow E,\ C \rightarrow F,\ F \rightarrow G$$

  si generano gli schemi R1(ABCDE), R2(CF), R3(FG) .

- Se 2 o più determinanti si determinano reciprocamente, si fondono gli schemi (più chiavi alternative per lo stesso schema).

  Esempio: se le FD su R(ABCD) sono: $A \rightarrow BC$, $B \rightarrow A$, $C \rightarrow D$

  si generano gli schemi R1(ABC), R2(CD) con B chiave in R1.

- Alla fine si verifica che esista uno schema la cui chiave è anche chiave dello schema originario (se non esiste lo si crea).

  Esempio: se le FD su R(ABCD) sono: $A \rightarrow C$, $B \rightarrow D$

  si generano gli schemi R1(AC), R2(BD), R3(AB) .

- N.B. L'algoritmo prevede un passo preliminare che consiste nel minimizzare l'insieme delle FD altrimenti non è garantita la correttezza del risultato.

Una limitazione non superabile

- In funzione del pattern di FD può non essere possibile decomporre in BCNF e preservare al tempo stesso le FD.

Dirigente	Agenzia	Sede
Rossetti	Spazio	Roma
Verdoni	Fast	Milano
Verdoni	Spazio	Milano
Moretti	Centrale	Milano
Moretti	Service	Milano

Agenzia, Sede → Dirigente:
ogni agenzia ha uno o più dirigenti (in questo caso in diverse sedi) e ogni dirigente può essere responsabile di più agenzie, però per ogni sede un'agenzia ha un solo dirigente responsabile.

Dirigente → Sede:
ogni dirigente opera in una sola sede.

- Agenzia, Sede → Dirigente: coinvolge tutti gli attributi e quindi nessuna decomposizione può preservare questa dipendenza!

Decomposizione dello schema

- Decomposizione in BCNF per (Dirigente, Agenzia, Sede) , con FD:

  Agenzia, Sede → Dirigente
  Dirigente → Sede

È innanzitutto opportuno osservare che {Agenzia, Dirigente} è una chiave
La decomposizione:

non è corretta perché è con perdita.

AGENZIE_SEDI

Agenzia	Sede
Spazio	Roma
Spazio	Milano
Fast	Milano
Centrale	Milano
Service	Milano

DIRIGENTI

Dirigente	Sede
Rossetti	Roma
Verdoni	Milano
Moretti	Milano

La decomposizione corretta è:
ma occorre una query di verifica per la FD
Agenzia, Sede → Dirigente

AGENZIE_DIRIGENTI

Agenzia	Dirigente
Spazio	Rossetti
Spazio	Verdoni
Fast	Verdoni
Centrale	Moretti
Service	Moretti

DIRIGENTI

Dirigente	Sede
Rossetti	Roma
Verdoni	Milano
Moretti	Milano

Riepilogo forme normali

- Prima Forma Normale (1NF)
  - Uno schema R(T) è in 1NF se e solo se il dominio di ciascun attributo comprende solo valori atomici (semplici, indivisibili) e il valore di ciascun attributo in una tupla è un valore singolo del dominio di quell'attributo.

- Seconda Forma Normale (2NF)
  - Uno schema R(T) con vincoli F è in 2NF se e solo se ogni attributo non-primo dipende completamente (non parzialmente) da ogni chiave candidata dello schema.

- Terza Forma Normale (3NF)
  - Uno schema R(T) con vincoli F è in 3NF se e solo se ogni attributo non-primo non dipende transitivamente da nessuna chiave.

- Forma Normale di Boyce-Codd (BCNF)
  - Uno schema R(T) con vincoli F è in BCNF se, per ogni dipendenza funzionale (non banale) $X \rightarrow Y$ definita su di esso, X è una superchiave di R(T).

Approccio nella pratica

- Se la relazione non è normalizzata si decompone in 3NF.
  - È sempre possibile con un algoritmo di sintesi ottenere decomposizioni in 3NF che sono senza perdita e preservano tutte le dipendenze.
- Si verifica se lo schema ottenuto è anche in BCNF; si noti che se una relazione ha una sola chiave allora le due forme normali coincidono.
  - Nella maggior parte dei casi pratici si può raggiungere l'obiettivo di una buona decomposizione in BCNF.
  - Esiste un algoritmo di sintesi in BCNF ma è di elevata complessità computazionale e genera un numero di schemi sovrabbondante.
- Se uno schema non è in BCNF si hanno tre alternative:
  - si lascia lo schema ottenuto così com'è, gestendo le anomalie residue, se l'applicazione lo consente;
  - si decompone in BCNF, predisponendo opportuni trigger o query di verifica;
  - si cerca di rimodellare la situazione iniziale, al fine di permettere di ottenere schemi BCNF.

Esercizio

Si dica in quale forma normale è lo schema:

ESAMI (Studente, Corso, Esaminatore, DataEsame, Voto)

nell'ipotesi che un esaminatore possa fare esami per un solo corso e, nel caso in cui non sia in BCNF, si determini un insieme di schemi normalizzati equivalenti in BCNF.

SOLUZIONE

Il requisito secondo cui un esaminatore svolge esami per un solo corso si traduce nella dipendenza funzionale

Esaminatore → Corso

Lo schema è in 3NF poiché Corso è un attributo primo (cioè è parte di una chiave).

… continua

Esercizio: soluzione

Lo schema non è in forma normale di Boyce-Codd poiché Esaminatore non è superchiave.

Una possibile soluzione è costituita dagli schemi:

ESAMI (Studente, Esaminatore, DataEsame,Voto)
ESAMINATORI_CORSI (Esaminatore, Corso)

Le due relazioni ottenute sono normalizzate in BCNF; risulta però impossibile verificare la dipendenza funzionale:

$$\text{Studente, Corso} \rightarrow \text{Esaminatore}$$

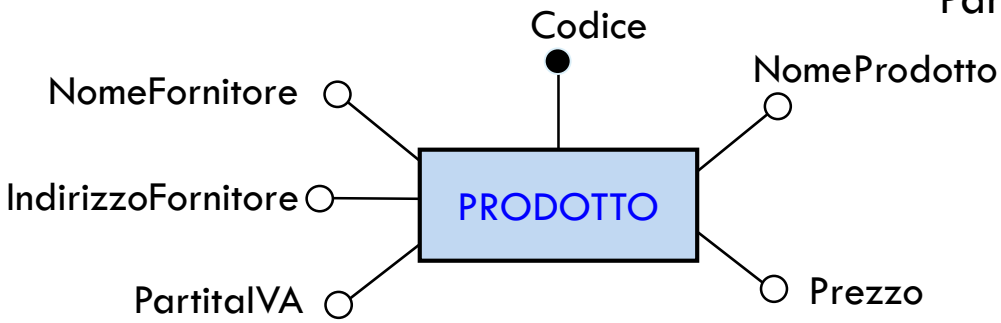
su una qualsiasi delle due relazioni. In altri termini, l'inserimento di due esami sostenuti dallo stesso studente per lo stesso corso con esaminatori diversi non può essere evitato a meno di fare riferimento contemporaneamente a entrambe le relazioni.

Progettazione e normalizzazione

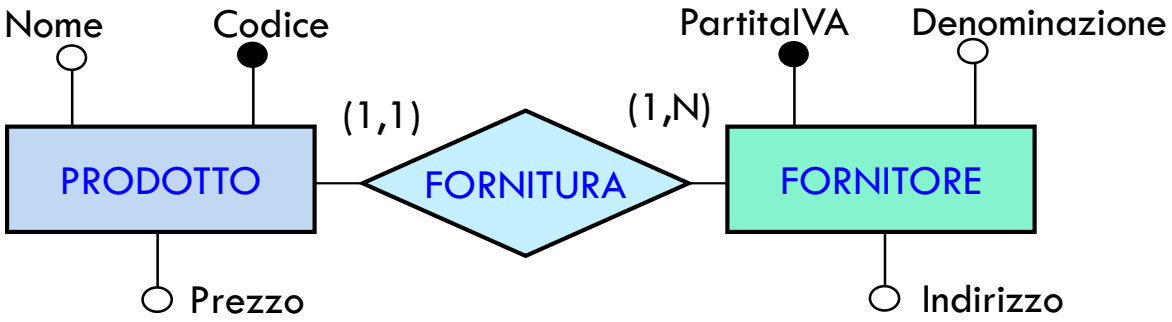
- La teoria della normalizzazione può essere usata nella progettazione logica per verificare lo schema relazionale finale.
- Si può usare anche durante la progettazione concettuale per verificare la qualità dello schema concettuale.

Esempio:

PartitaIVA → NomeFornitore, IndirizzoFornitore

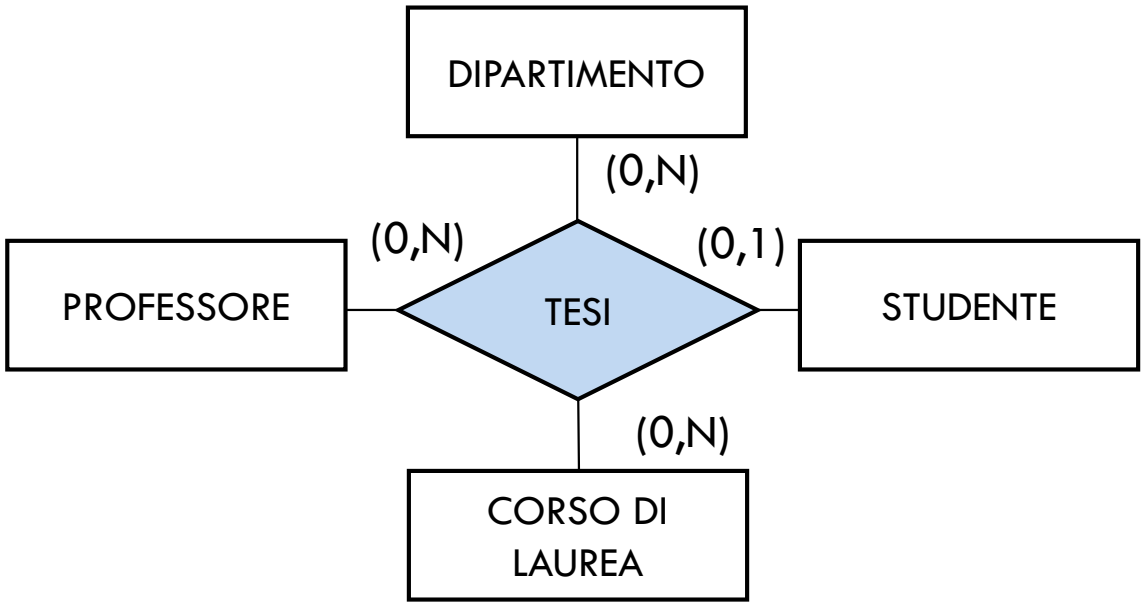


Sfruttando la FD, e modificando i nomi di alcuni attributi, si perviene a:



Analisi di associazioni n-arie (1)

Le associazioni n-arie spesso nascondono FD che possono dar luogo a schemi non normalizzati.



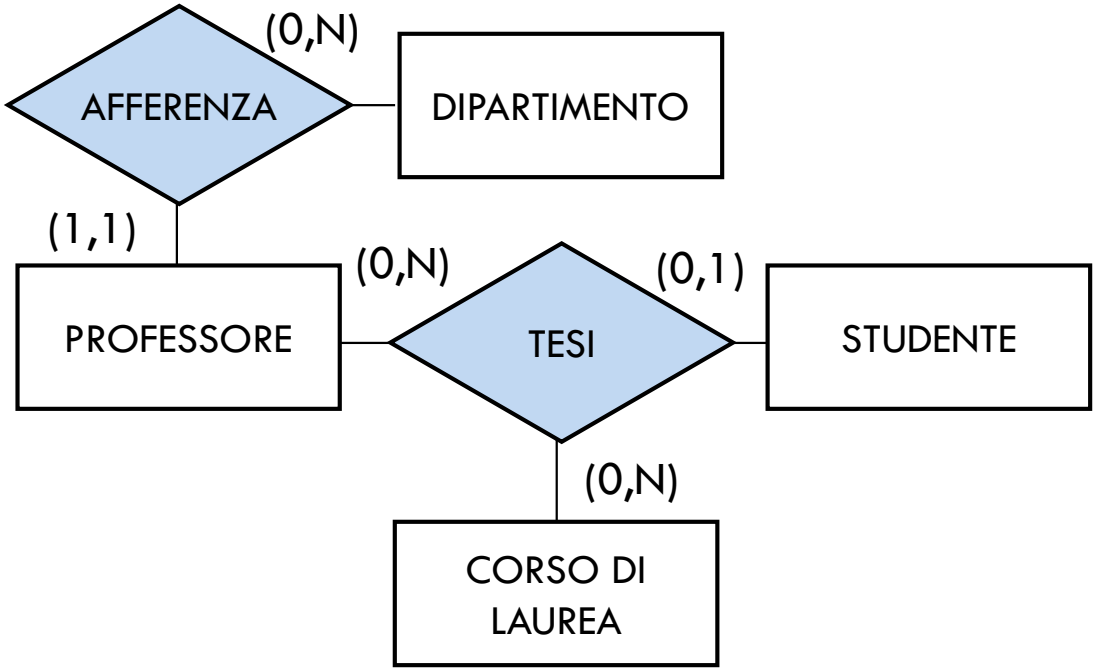
Studente → CorsoDiLaurea
Studente → Professore
Professore → Dipartimento

TESI(Studente, Professore, Dipartimento, CorsoDiLaurea)

non è in 3NF a causa della dipendenza transitiva Professore → Dipartimento

Analisi di associazioni n-arie (2)

Si ristruttura lo schema di conseguenza:



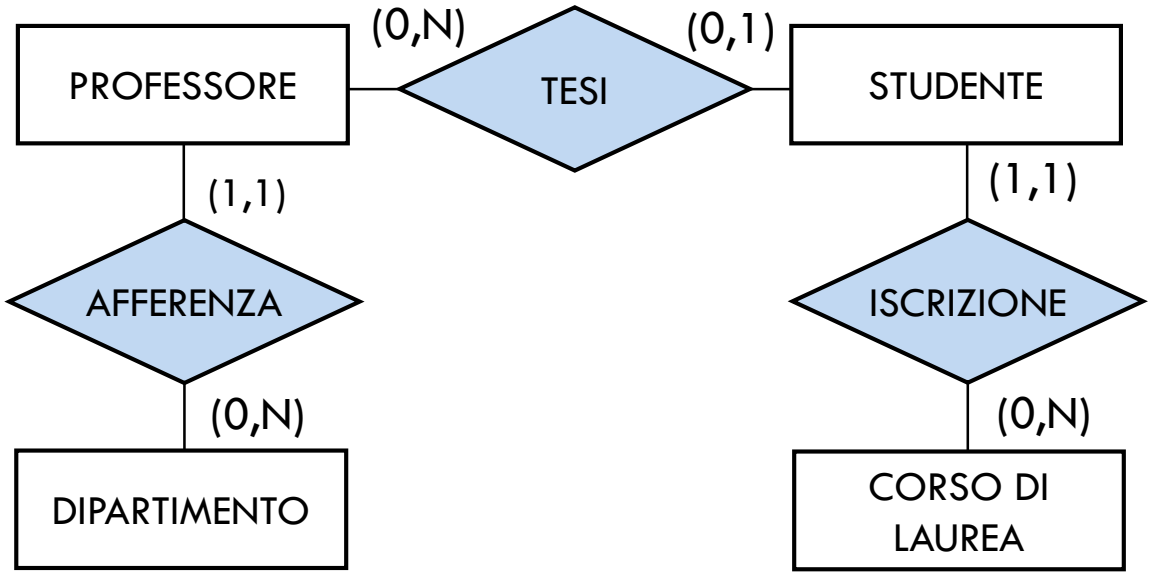
TESI(<u>Studente</u>, Professore, CorsoDiLaurea) è ora in BCNF.

Analisi di associazioni n-arie (3)

L'associazione TESI in realtà include 2 FD, tra loro indipendenti:

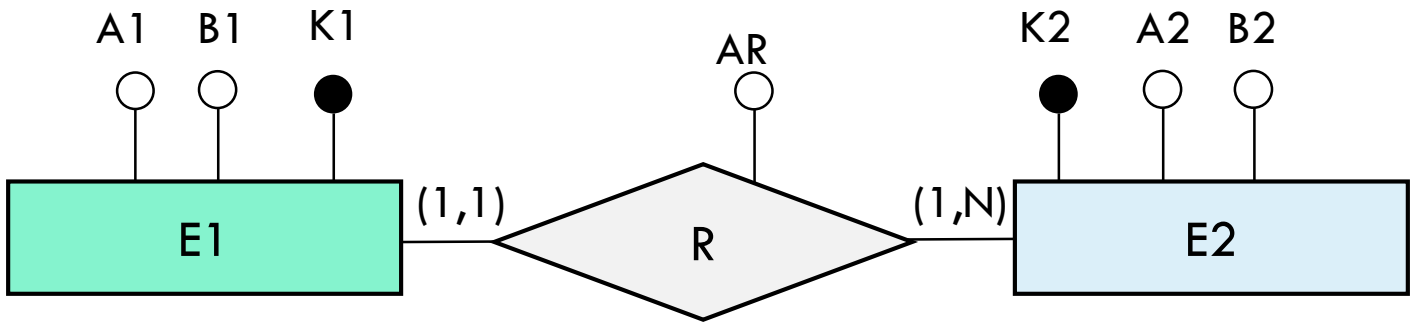
- Studente → CorsoDiLaurea (iscrizione)
- Studente → Professore (per chi ha un relatore)

- È quindi opportuno procedere a un'ulteriore ristrutturazione:



FD e modello E/R

- È bene abituarsi a "leggere" uno schema E/R anche in termini di FD.
- A tal fine si considerano le cardinalità massime delle associazioni:

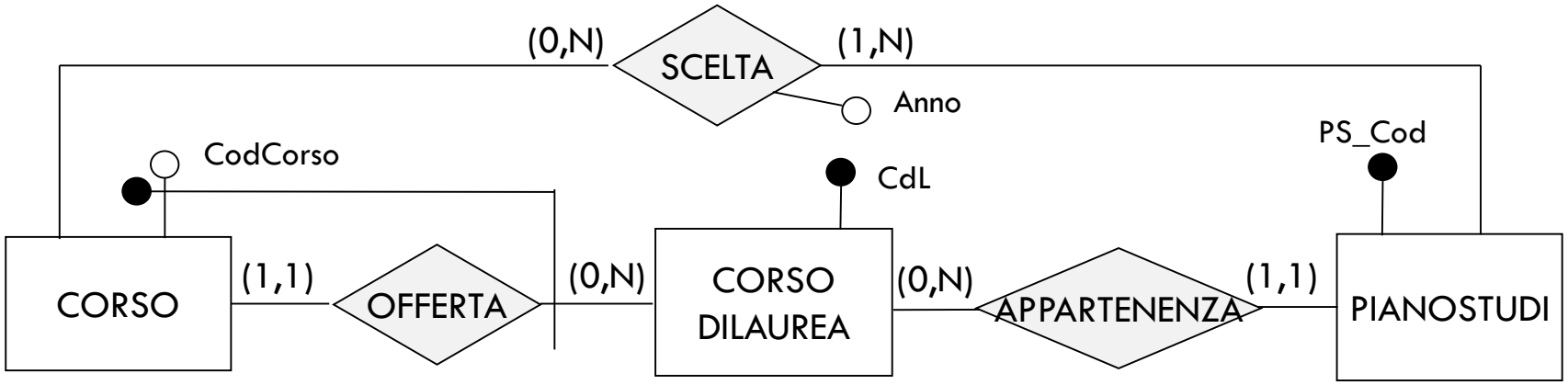


K1 → A1, B1
K2 → A2, B2
K1 → K2, AR poiché max-card(E1,R) = 1

- Si suggerisce di rivedere le regole per la traduzione delle associazioni in termini di FD tra gli identificatori delle entità e di normalizzazione degli schemi.

Possiamo fare a meno delle FD?

Anche se in molti casi una buona progettazione concettuale rende superfluo ragionare in termini di FD, vi sono schemi E/R "corretti" che danno luogo a schemi relazionali non normalizzati.



Vincolo: si possono scegliere solo corsi offerti dal proprio CdL.

La traduzione dell'associazione SCELTA genera lo schema:

SCELTE(CdL, CodCorso, PS_Cod, Anno)

individuando solo una superchiave; la vera chiave è {CodCorso, PS_Cod}. Dunque a causa della dipendenza parziale PS_Cod → CdL lo schema non è in 2NF. La traduzione corretta dell'associazione in 3NF è:

SCELTE(CodCorso, PS_Cod, Anno)

È sempre opportuno normalizzare?

- La normalizzazione non deve essere intesa come un obbligo; infatti in alcune situazioni le anomalie che si riscontrano in schemi non normalizzati sono un male minore rispetto alla situazione che si verrebbe a creare normalizzando.

- In particolare, gli aspetti da considerare sono:
  - normalizzare elimina le anomalie ma può appesantire l'esecuzione di certe operazioni (join tra gli schemi normalizzati);
  - la frequenza con cui i dati sono soggetti a modifica incide su qual è la scelta più opportuna (relazioni "quasi statiche" danno un minor numero di problemi se non sono normalizzate);
  - la ridondanza presente in relazioni non normalizzate va quantificata al fine di comprendere quanto possa incidere sull'occupazione di memoria e sui costi derivanti dall'aggiornamento di repliche di una stessa informazione.

Domande?