

Basi di Dati

Il modello relazionale

Corso B

Base dati di esempio: Ricoveri Ospedalieri

Reparto di Pediatria Certificato di ricovero

Torino 7/10/2014

Si attesta che il bambino Gino Rossetti, residente in provincia di Asti, è stato ricoverato il 5/10/2014.

In fede
Il Primario
Prof. Mario Bisi

PAZIENTI

<u>COD</u>	Cognome	Nome	Residenza	AnnoNascita
A102	Necchi	Luca	TO	1950
B372	Rossigni	Piero	NO	1940
B543	Missoni	Nadia	TO	1960
B444	Missoni	Luigi	VC	2000
S555	Rossetti	Gino	AT	2010

REPARTI

<u>COD</u>	Nome-Rep	Primario
A	Chirurgia	203
B	Pediatria	574
C	Medicina	530
L	Lab-Analisi	530
R	Radiologia	405

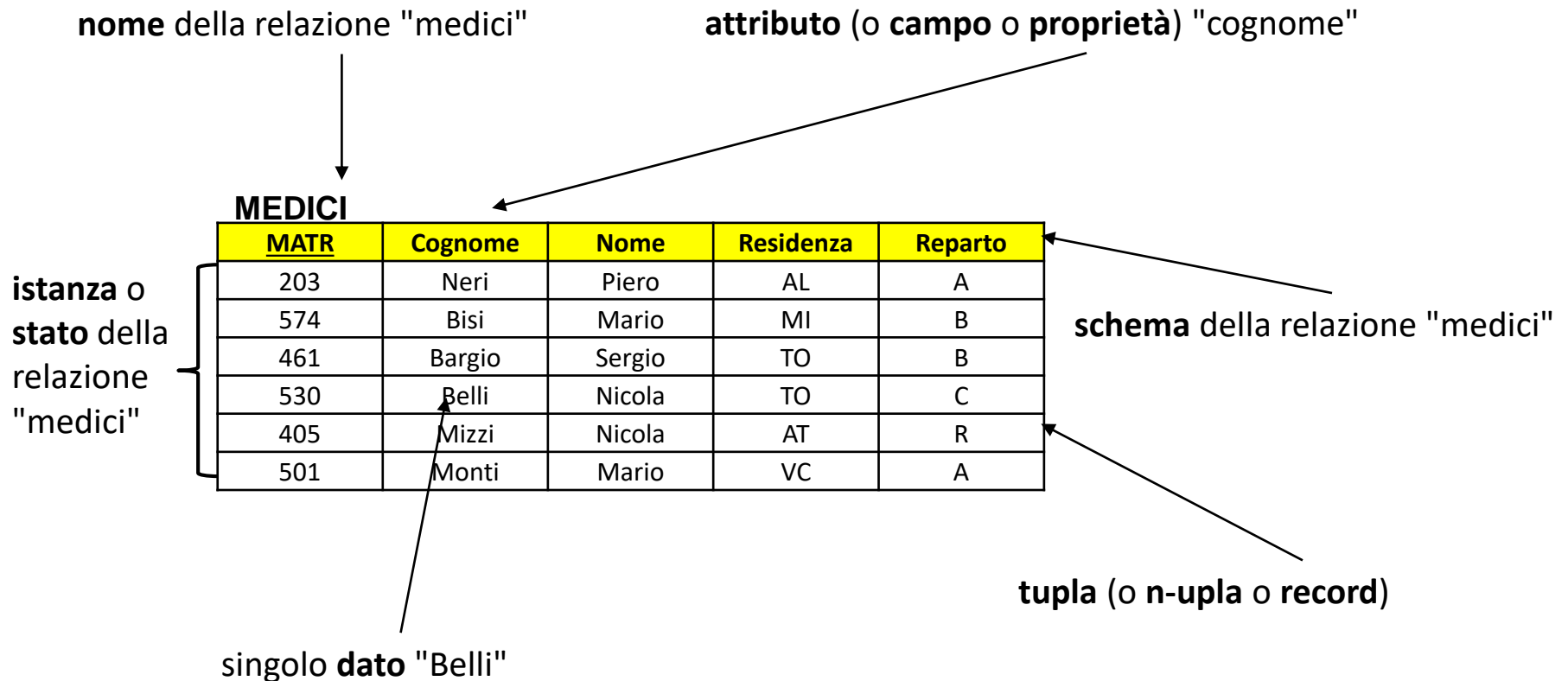
RICOVERI

<u>PAZ</u>	Inizio	Fine	Reparto
A102	2/05/2014	9/05/2014	A
A102	2/12/2004	2/01/2005	A
S555	5/10/2014	3/12/2014	B
B444	1/12/2004	2/01/2005	B
S555	6/09/2015	1/11/2015	A

MEDICI

<u>MATR</u>	Cognome	Nome	Residenza	Reparto
203	Neri	Piero	AL	A
574	Bisi	Mario	MI	B
461	Bargio	Sergio	TO	B
530	Belli	Nicola	TO	C
405	Mizzi	Nicola	AT	R
501	Monti	Mario	VC	A

Esempio: relazione



Lo schema fornisce l'interpretazione dei dati → trasforma i dati in informazioni

Esempio: relazioni tra insiemi omogenei

Istanze dell'insieme "reparti" in relazione con istanze dell'insieme "medici" tramite il valore dell'attributo "primario"

REPARTI

<u>COD</u>	Nome-Rep	Primario
A	Chirurgia	203
B	Pediatria	574
C	Medicina	530
L	Lab-Analisi	530
R	Radiologia	405

Attributo identificatore della relazione "medici"

MEDICI

<u>MATR</u>	Cognome	Nome	Residenza	Reparto
203	Neri	Piero	AL	A
574	Bisi	Mario	MI	B
461	Bargio	Sergio	TO	B
530	Belli	Nicola	TO	C
405	Mizzi	Nicola	AT	R
501	Monti	Mario	VC	A

1. Relazione (definizione)

Una **relazione** è definita

- da uno **schema** [di una relazione]
- dalle **istanze** della relazione (o **stato** della relazione)

2. Attributo (definizione)

L'**attributo** di una relazione (attributo relazionale) è definito come una coppia $A_i : T_i$ dove

- T_i è il **tipo** dell'attributo
- A_i è il **nome** dell'attributo
- Tipi standard (o predefiniti):
 - Integer, Real, String, Char, Date, Boolean
- Tipi utente (tipi di attributi definiti dall'utente)
 - Esempio: tipo enumerativo (lista di città)

Tipo di un attributo

Tipo T_i caratterizzato da

- T_i : **nome** identificativo del tipo
(ad esempio: Integer, Real, ecc...)
- D_i : **dominio** di valori
- collezione di **operazioni** abilitate ad agire su D_i
(ad esempio: +, -, x, /...)
- operazioni di **confronto** su D_i
(ad esempio: <, <=, >=, =, ...)

Dominio di un attributo

Un tipo è **sempre associato** ad un dominio, ovvero ad un insieme (potenzialmente infinito) di valori

Possiamo immaginare una funzione *dom* che associa all'attributo A_i (di tipo T_i) il suo dominio D_i , ovvero:

$$D_i = dom (A_i)$$

Valore nullo

Cosa succede se, nel momento in cui inserisco un paziente non ne conosco l'anno di nascita?

E' possibile inserire delle tuple con attributi mancanti in una relazione?

E' previsto un valore speciale, chiamato **valore nullo** o **NULL** (un vero e proprio valore) con la seguente proprietà generale

$$\forall T_i, \text{ NULL } \in D_i$$

ovvero, il valore NULL appartiene al dominio D_i di qualsiasi tipo T_i

Valore nullo: esempio

PAZIENTI

<u>COD</u>	Cognome	Nome	Residenza	AnnoNascita
A102	Necchi	Luca	TO	1950
B372	Rossigni	Piero	NO	1940
B543	Missoni	Nadia	TO	1960
B444	Missoni	Luigi	VC	2000
S555	Rossetti	Gino	AT	2010
Z999	Rossi	Giovanni	CN	NULL



3. Schema di una relazione

Lo **schema di una relazione** è un **insieme di attributi** con la seguente notazione:

$$\{ A_1 : T_1, A_2 : T_2, \dots, A_n : T_n \}$$

(schema di relazione **senza nome** composto solo dall'insieme degli attributi)

N.B.:

- Insieme di attributi $\{ A_i : T_i \}$ tutti con **nomi distinti**, cioè

$$\forall i, j \ A_i \neq A_j$$

- possono esistere invece **tipi uguali** all'interno della stessa relazione
- l'ordine degli attributi è **irrilevante** nella definizione formale di Codd

Schemi con nomi

Lo schema di una relazione **con nome** ha la seguente notazione:

$$R(A_1 : T_1 , A_2 : T_2 , \dots , A_n : T_n)$$

dove R è un generico nome di relazione (ad esempio PAZIENTI)

Notazione semplificata

Spesso, per semplificare, si sottintende il tipo:

- $\{ A_1 : T_1, A_2 : T_2, \dots, A_n : T_n \} \rightarrow \{ A_1, A_2, \dots, A_n \}$
- $R(A_1 : T_1, A_2 : T_2, \dots, A_n : T_n) \rightarrow R(A_1, A_2, \dots, A_n)$

Nei DBMS, però, il tipo va **sempre** specificato

Esempio

- *PAZIENTI(COD, Cognome, Nome, Residenza, AnnoNascita)*

Notazione generale

Definendo l'insieme di attributi A come

$$A: \{ A_1 : T_1, A_2 : T_2, \dots, A_n : T_n \}$$

posso utilizzare la notazione

$$R(A)$$

dove R è il nome dello schema

Grado di una relazione

La **cardinalità** $|A|$ di uno schema di relazione è il numero di attributi dello schema A , ed è detto **grado** della relazione

N.B.:

- L'insieme di attributi di uno schema di una relazione è sempre un insieme **non vuoto**

$$|A| \geq 1$$

4. Istanza (o stato) della relazione

Dato uno schema

$$A: \{ A_1 : T_1, A_2 : T_2, \dots, A_n : T_n \}$$

(per semplicità consideriamo l'insieme A **ordinato**)

l'**istanza**, o **stato** r di una relazione è definito come un insieme di tuple

$$\langle v_1, v_2, \dots, v_n \rangle$$

per cui vale la seguente proprietà:

$$\forall i, v_i \in \text{dom} (A_i)$$

Cardinalità di una relazione

La **cardinalità** $|r|$ dell'istanza di una relazione r è il numero di tuple appartenenti alla relazione r , ed è detta semplicemente **cardinalità della relazione**

N.B.:

- Contrariamente alla relazione, l'istanza di una relazione **può essere vuota**, ovvero

$$|r| \geq 0$$

Notazioni ibride (compatte)

Utilizzo lettere **maiuscole** per le **relazioni**:

R oppure *PAZIENTI*

Utilizzo lettere **minuscole** per l'**istanza** delle relazioni:

r oppure *pazienti*

Specifico lo schema anche per le istanze:

- $r(A_1, A_2, \dots, A_n)$
- *pazienti(COD, Cognome, Nome, Residenza, AnnoNascita)*

Altre: $r(R)$, *pazienti(PAZIENTI)* (poco usate), *r* (se lo schema *R* è ben chiaro)

Notazione tabellare

Schema (insieme di attributi)

<u>COD</u>	Cognome	Nome	Residenza	AnnoNascita
A102	Necchi	Luca	TO	1950
B372	Rossigni	Piero	NO	1940
B543	Missoni	Nadia	TO	1960
B444	Missoni	Luigi	VC	2000
S555	Rossetti	Gino	AT	2010

Istanza (insieme di tuple)

5. Relazione (notazione)

La relazione è definita da una **coppia**

$$\langle R, r \rangle$$

dove R è lo schema della relazione, ed r è l'istanza o stato della relazione

6. Tupla (notazione)

Data la relazione $r(A_1, A_2, \dots, A_n)$, per indicare una singola **tupla** si utilizza la notazione

$$t = \langle v_1, v_2, \dots, v_n \rangle$$

t prende valori all'interno della relazione R quindi

$$t \in r(A_1, A_2, \dots, A_n)$$

Valori di una tupla

Per indicare il valore di una tupla t in corrispondenza dell'attributo A_i si utilizza la notazione $t[A_i]$, quindi

$$t[A_i] = v_i$$

Per indicare i valori di una tupla t in corrispondenza di più attributi A_i, A_j, A_k , si utilizza la notazione $t[A_i, A_j, A_k]$, quindi

$$t[A_i, A_j, A_k] = \langle v_i, v_j, v_k \rangle$$

Per richiamare la tupla su tutti gli attributi si utilizza t

Altre notazioni: $t.A_i$, $t.(A_i, A_j, A_k)$, $t[A]$, $t.A$

Relazione matematica vs. relazione di Codd

In matematica la relazione s è un sottoinsieme del prodotto cartesiano di due o più insiemi

$$s \subseteq \text{dom}(A_1) \times \text{dom}(A_2) \times \dots \times \text{dom}(A_n)$$

DIFFERENZE:

- In matematica $A \times B$ è **diverso** da $B \times A$, ovvero le relazioni sono strettamente legate all'ordine degli insiemi
- Per il modello relazionale, l'ordine degli attributi dello schema è **irrilevante**

Tupla come rappresentazione tabellare

- Non essendo importante l'ordine degli attributi, la tupla non è altro che una **funzione** che fa corrispondere ad ogni attributo un valore del suo dominio

$$t: A \rightarrow D$$

A	D
COD	A102
Cognome	Necchi
Nome	Luca
Residenza	TO
AnnoNascita	1950

7. Istanza di una relazione

L'istanza r di una relazione è dunque un insieme di funzioni $\{ t_1, t_2, \dots, t_n \}$, ognuna delle quali definisce una particolare corrispondenza tra attributi e valori

Le funzioni t_i sono tutte distinte tra loro (in una relazione non può esistere una tupla identica ad un'altra)

Funzione totale

Una **funzione totale** $f: X \rightarrow Y$ fa corrispondere ad **ogni** valore dell'insieme X un valore dell'insieme Y

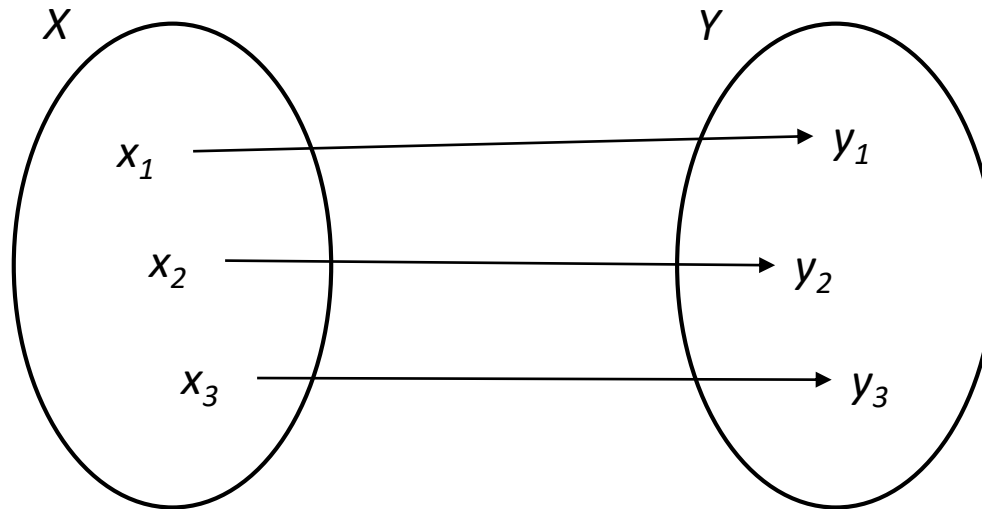


Diagramma di Venn

Tupla compatibile con uno schema

La tupla t è compatibile con lo schema A di una relazione se

- t è una **funzione totale** su A
- ogni valore prodotto da t appartiene a D
- vale il vincolo

$$\forall A_i, t[A_i] \in \text{dom}(A_i)$$

8. Vincoli locali

I **vincoli locali** sono caratterizzazioni dei valori possibili delle tuple in modo che tali valori rispettino (non siano in contrasto con) la realtà che vogliamo rappresentare

Vincolo locale di dominio

Per ogni attributo ho un insieme **ben definito** (anche infinito) di valori possibili

N.B.:

- Qualsiasi valore v del dominio D_i è atomico
- Un attributo il cui valore è una relazione non è atomico
 - Alcuni DBMS lo permettono (ad esempio Oracle)

Relazione in prima forma normale

Una relazione è **in prima forma normale** quando tutti gli attributi sono abbinati a domini di **valori atomici**

- Integer
- Boolean
- Real
- Date
- String
- Char
- Tipi utente semplici (enumerazioni)

Vincolo sui valori nulli

Un vincolo NOT NULL su un attributo A_i è soddisfatto se e solo se per ogni $t \in r$, il valore $t[A_i]$ **non è nullo**

Funzionamento **pratico**:

Al momento del caricamento dei dati in una tabella, il DBMS verifica

1. che la tupla sia compatibile
2. che i valori degli attributi per cui è attivo il vincolo NOT NULL siano non nulli.

Altri vincoli

- Restrizione sui domini
 - Una persona non può avere più di 120 anni, e non può essere nata nel futuro
 - Una temperatura corporea di un paziente non può superare i 42 gradi ed essere inferiore ai 34
- Confronti sui domini
 - La data di inizio ricovero dev'essere antecedente alla data di fine ricovero

I DBMS forniscono la sintassi necessaria per la verifica di questi vincoli

Vincolo di identificazione

- L'identificazione nel modello relazionale si attua attraverso il vincolo di **chiave relazionale**
- E' il contesto applicativo che suggerisce la nozione di vincolo di chiave relazionale (ad esempio COD dei pazienti o MATR dei medici)
- Le chiavi relazionali possono essere composte da più valori (es. tabella RICOVERI)
- Nella relazione di Codd gli elementi (tuple) dell'insieme (istanza) r devono essere tutti distinti tra di loro

Definizione di superchiave

Per poter definire in maniera formale il vincolo di chiave relazionale dobbiamo introdurre prima la nozione di superchiave

Data una relazione $r(A)$, un sottoinsieme di attributi $sk \subseteq A$ è una **superchiave** se:

$$\forall i, j (t_i[sk] = t_j[sk]) \rightarrow (t_i[A] = t_j[A])$$

Esempio: nella tabella PAZIENTI, (COD, Nome) è una superchiave

Verifica dell'implicazione

Ricordiamo che l'**implicazione logica**

$$\alpha \rightarrow \beta \text{ equivalente a } (\neg\alpha \vee \beta)$$

è vera

- sempre se α è falsa
- se α è vera e β è vera

è falsa invece se α è vera e β è falsa

Verifica dell'implicazione

$sk = \{ COD, Cognome \}$ è una superchiave?

 **PAZIENTI**

<u>COD</u>	Cognome	Nome	Residenza	AnnoNascita	
A102	Necchi	Luca	TO	1950	t_1
B372	Rossigni	Piero	NO	1940	t_2
B543	Missoni	Nadia	TO	1960	t_3
B444	Missoni	Luigi	VC	2000	t_4
S555	Rossetti	Gino	AT	2010	t_5

$t_1[\{COD, Cognome\}] = t_1[\{COD, Cognome\}]$ e $t_1 = t_1$
 $t_1[\{COD, Cognome\}] \neq t_2[\{COD, Cognome\}]$ e $t_1 \neq t_2$
 $t_1[\{COD, Cognome\}] \neq t_3[\{COD, Cognome\}]$ e $t_1 \neq t_3$
 $t_1[\{COD, Cognome\}] \neq t_4[\{COD, Cognome\}]$ e $t_1 \neq t_4$
 $t_1[\{COD, Cognome\}] \neq t_5[\{COD, Cognome\}]$ e $t_1 \neq t_5$

Verifica dell'implicazione

$sk = \{ COD, Cognome \}$ è una superchiave?

PAZIENTI



<u>COD</u>	Cognome	Nome	Residenza	AnnoNascita
A102	Necchi	Luca	TO	1950
B372	Rossigni	Piero	NO	1940
B543	Missoni	Nadia	TO	1960
B444	Missoni	Luigi	VC	2000
S555	Rossetti	Gino	AT	2010

t_1
 t_2
 t_3
 t_4
 t_5

$t_2[\{COD, Cognome\}] = t_2[\{COD, Cognome\}]$ e $t_2 = t_2$

$t_2[\{COD, Cognome\}] \neq t_3[\{COD, Cognome\}]$ e $t_2 \neq t_3$

$t_2[\{COD, Cognome\}] \neq t_4[\{COD, Cognome\}]$ e $t_2 \neq t_4$

$t_2[\{COD, Cognome\}] \neq t_5[\{COD, Cognome\}]$ e $t_2 \neq t_5$

Verifica dell'implicazione

$sk = \{ COD, Cognome \}$ è una superchiave?

PAZIENTI



<u>COD</u>	Cognome	Nome	Residenza	AnnoNascita	
A102	Necchi	Luca	TO	1950	t_1
B372	Rossigni	Piero	NO	1940	t_2
B543	Missoni	Nadia	TO	1960	t_3
B444	Missoni	Luigi	VC	2000	t_4
S555	Rossetti	Gino	AT	2010	t_5

$t_3[\{COD, Cognome\}] = t_3[\{COD, Cognome\}]$ e $t_3 = t_3$

$t_3[\{COD, Cognome\}] \neq t_4[\{COD, Cognome\}]$ e $t_3 \neq t_4$

$t_3[\{COD, Cognome\}] \neq t_5[\{COD, Cognome\}]$ e $t_3 \neq t_5$

Verifica dell'implicazione

$sk = \{ COD, Cognome \}$ è una superchiave?

PAZIENTI

<u>COD</u>	Cognome	Nome	Residenza	AnnoNascita	
A102	Necchi	Luca	TO	1950	t_1
B372	Rossigni	Piero	NO	1940	t_2
B543	Missoni	Nadia	TO	1960	t_3
B444	Missoni	Luigi	VC	2000	t_4
S555	Rossetti	Gino	AT	2010	t_5



$t_4[\{COD, Cognome\}] = t_4[\{COD, Cognome\}]$ e $t_4 = t_4$

$t_4[\{COD, Cognome\}] \neq t_5[\{COD, Cognome\}]$ e $t_4 \neq t_5$

Verifica dell'implicazione

$sk = \{ COD, Cognome \}$ è una superchiave?

PAZIENTI

<u>COD</u>	Cognome	Nome	Residenza	AnnoNascita	
A102	Necchi	Luca	TO	1950	t_1
B372	Rossigni	Piero	NO	1940	t_2
B543	Missoni	Nadia	TO	1960	t_3
B444	Missoni	Luigi	VC	2000	t_4
S555	Rossetti	Gino	AT	2010	t_5



$$t_5[\{COD, Cognome\}] = t_5[\{COD, Cognome\}] \text{ e } t_5 = t_5$$

la proprietà è verificata **per ogni** coppia di tuple

Istanza diversa

$sk = \{ COD, Cognome \}$ è una superchiave?

PAZIENTI					
<u>COD</u>	Cognome	Nome	Residenza	AnnoNascita	
A102	Necchi	Luca	TO	1950	t_1
B372	Rossigni	Piero	NO	1940	t_2
B543	Missoni	Nadia	TO	1960	t_3
B444	Missoni	Luigi	VC	2000	t_4
S555	Rossetti	Gino	AT	2010	t_5
S555	Rossetti	Aldo	CN	2009	t_6

per la coppia (t_5, t_6) , è vero l'antecedente **ma non** il conseguente cioè
 $(t_i[sk] = t_j[sk]) \not\rightarrow (t_i[A] = t_j[A])$

Quantificazione universale \rightarrow esistenziale

Ricordiamo che l'**implicazione logica**

$$\alpha \rightarrow \beta \text{ equivalente a } (\neg\alpha \vee \beta)$$

neghiamo

$$\neg(\alpha \rightarrow \beta) \text{ equivalente a } \neg(\neg\alpha \vee \beta) \text{ cioè } (\alpha \wedge \neg\beta)$$

e con la doppia negazione (e la quantificazione esistenziale):

$$\neg \forall i, j (t_i[sk] = t_j[sk]) \rightarrow (t_i \neq t_j)$$

Chiave candidata

$k \subseteq A$ è **chiave candidata** se

- k è **superchiave** di R
- k è **superchiave minimale** (ogni sottoinsieme proprio di attributi di k non deve soddisfare la condizione di superchiave)

Esempio: COD è superchiave, quindi {COD, Cognome} non è chiave candidata

Esempio

RICOVERI

PAZ	Inizio	Fine	Reparto
A102	2/05/2014	9/05/2014	A
A102	2/12/2004	2/01/2005	A
S555	5/10/2014	3/12/2014	B
B444	2/12/2004	2/01/2005	B
S555	6/09/2015	1/11/2015	A

- $\{PAZ, Inizio, Fine\}$ è una superchiave
- $\{PAZ, Inizio\}$ soddisfa la proprietà di superchiave (non esistono tuple distinte con stesso paziente e stessa data di inizio ricovero)
- $\{PAZ, Inizio\}$ è minimale?
 - Sottoinsieme proprio $\{PAZ\}$ non è superchiave
 - Sottoinsieme proprio $\{Inizio\}$ non è superchiave $\{PAZ, Inizio\}$ è chiave candidata

Proprietà delle superchiavi

Se sk è una superchiave allora $sk \subseteq w \subseteq A$ è una superchiave

quindi:

- A è necessariamente una superchiave (per la definizione stessa di istanza, ogni tupla è distinta)
- A è superchiave, quindi A deve contenere sempre almeno una chiave candidata

Chiavi candidate

STUDENTI

<u>MATR</u>	Cognome	Nome	LuogoNascita	DataNascita	CodiceFiscale
102	Necchi	Luca	Torino	2/10/90	NCHLCU
372	Rossigni	Piero	Novara	2/12/91	RSGPRI
543	Missoni	Nadia	Torino	22/11/89	MSNDA
444	Missoni	Luigi	Vercelli	4/2/90	MSNLGU
555	Rossetti	Gino	Asti	4/2/90	RSTGNI

- $\{MATR\}$ è una superchiave
- $\{CodiceFiscale\}$ è anche una chiave
- $\{Cognome, Nome, LuogoNascita, DataNascita\}$ è anche una chiave (composta)

Il progettista deve sempre scegliere tra le chiavi candidate una **chiave principale** (o **chiave primaria**)

Chiave principale

La chiave principale $pk \subseteq A$ è una chiave scelta dal progettista tra tutte le possibili chiavi candidate

- Tutti gli attributi di una chiave principale devono essere **NOT NULL** (non nulli)
 - una chiave candidata può ammettere valori nulli

Nell'esempio "Ricoveri ospedalieri" le chiavi principali sono sottolineate

9. Correttezza di una relazione

Un'istanza di relazione $r(A)$ è corretta se ogni tupla $t \in r$ è compatibile con lo schema A e sono soddisfatti tutti i vincoli locali (o intrarelazionali)

10. Schema di una base dati

Lo schema di una base dati è un insieme di schemi di relazione tutti con nome

$$S = \{R_1, R_2, \dots, R_n\}$$

con le seguenti proprietà:

- i nomi devono essere tutti distinti
- ad ogni schema si abbinano i rispettivi vincoli locali
- ogni relazione R_i esibisce una chiave principale

Esempio di schema

RicoveriOspedalieri = {PAZIENTI, MEDICI, REPARTI, RICOVERI}

con

- PAZIENTI(COD, Cognome, Nome, Residenza, AnnoNascita)
- MEDICI(MATR, Cognome, Nome, Residenza, Reparto)
- REPARTI(COD, Nome, Primario)
- RICOVERI(PAZ, Inizio, Fine, Reparto)

Criterio di buon progetto

Criteri di buon progetto di uno schema di basi dati S

- in caso di attributi omonimi che si riferiscono al medesimo concetto, il tipo dev'essere lo stesso
- in caso di attributi omonimi in più relazioni, il concetto espresso dev'essere lo stesso
 - Esempio, concetto Cognome in MEDICI e PAZIENTI

quindi:

- evitare **omonimie** (stessi nomi per concetti diversi)
 - nome (di una persona o di un prodotto)
- **sinonimie** (nomi diversi per stessi concetti)
 - corso e insegnamento (per unità didattica)

Vincoli globali definiti su S

Così come è possibile definire dei vincoli locali (**intrarelazionali**) su ogni relazione R è altresì possibile definire vincoli globali (**interrelazionali**) su uno schema di basi di dati S

Il più importante è il **vincolo di integrità referenziale**

Vincolo di integrità referenziale

Consideriamo l'esempio

- PAZIENTI(COD, Cognome, Nome, Residenza, AnnoNascita)
- RICOVERI(PAZ, Inizio, Fine, Reparto)

PAZIENTI

<u>COD</u>	Cognome	Nome	Residenza	AnnoNascita
A102	Necchi	Luca	TO	1950
B372	Rossigni	Piero	NO	1940
B543	Missoni	Nadia	TO	1960
B444	Missoni	Luigi	VC	2000
S555	Rossetti	Gino	AT	2010

RICOVERI

<u>PAZ</u>	<u>Inizio</u>	<u>Fine</u>	<u>Reparto</u>
A102	2/05/2014	9/05/2014	A
A102	2/12/2004	2/01/2005	A
S555	5/10/2014	3/12/2014	B
B444	1/12/2004	2/01/2005	B
S555	6/09/2015	1/11/2015	A

Il valore 'A102' dell'attributo PAZ nella prima tupla di RICOVERI si riferisce al codice di un ben preciso paziente presente nella relazione PAZIENTI

Vincolo di integrità referenziale

Consideriamo l'esempio

- PAZIENTI(COD, Cognome, Nome, Residenza, AnnoNascita)
- RICOVERI(PAZ, Inizio, Fine, Reparto)

PAZIENTI

<u>COD</u>	Cognome	Nome	Residenza	AnnoNascita
A102	Necchi	Luca	TO	1950
B372	Rossigni	Piero	NO	1940
B543	Missoni	Nadia	TO	1960
B444	Missoni	Luigi	VC	2000
S555	Rossetti	Gino	AT	2010

RICOVERI

<u>PAZ</u>	<u>Inizio</u>	<u>Fine</u>	<u>Reparto</u>
A102	2/05/2014	9/05/2014	A
A102	2/12/2004	2/01/2005	A
S555	5/10/2014	3/12/2014	B
B444	1/12/2004	2/01/2005	B
S555	6/09/2015	1/11/2015	A
Z999	02/06/96	07/06/96	C

Il paziente relativo al valore Z999 non è presente in PAZIENTI, pur essendo la nuova tupla compatibile con lo schema RICOVERI
La base di dati non riflette correttamente la realtà in questione

Vincolo di integrità referenziale

Il progettista può inserire un vincolo di integrità referenziale che impone che

- se in RICOVERI c'è una tupla t con $t.[PAZ]=X$ allora, necessariamente in PAZIENTI dovrà esistere una tupla t con $t.[COD]=X$

Se il vincolo non è rispettato la base dati non è corretta

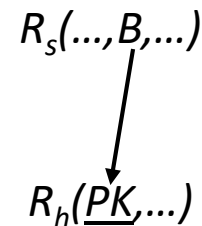
11. Vincolo di integrità referenziale

Consideriamo una relazione $R_h(\underline{PK}, \dots)$, dove PK è l'insieme di attributi della chiave principale, e una relazione $R_s(\dots, B, \dots)$ dove B è un insieme di attributi qualsiasi.

Esiste un **vincolo di integrità referenziale** degli attributi B rispetto alla tavola R_h se:

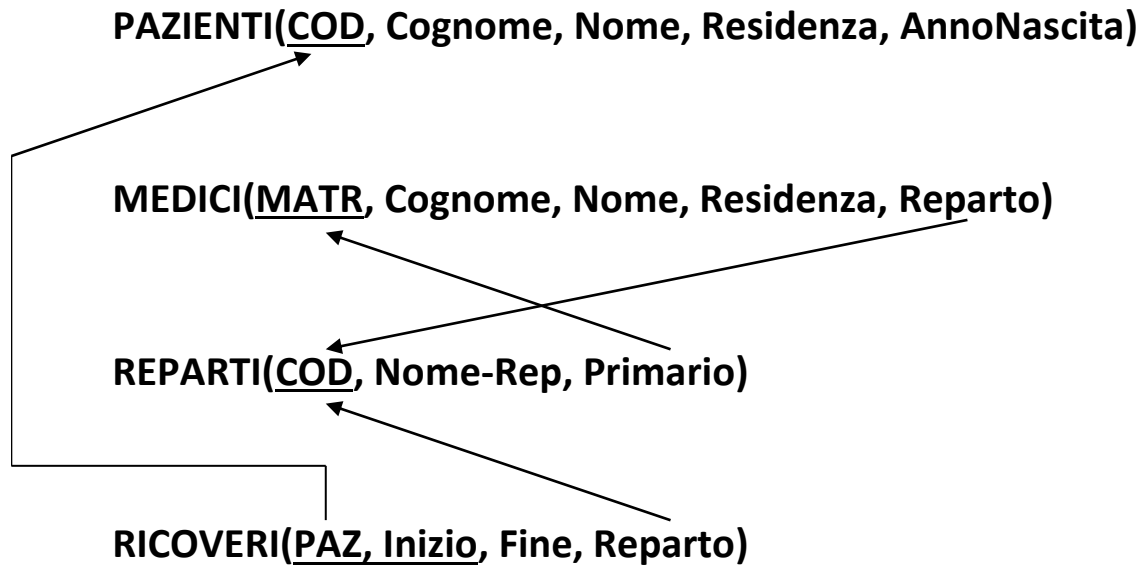
$$\forall t_i (t_i \in r(R_s)) \Rightarrow \exists t_j (t_j \in r(R_h)) \wedge (t_i[B] = t_j[PK])$$

Graficamente si indica così



Esempio

Schema relazionale con vincoli di integrità referenziali:



Condizione necessaria per il vincolo

E' necessario che i domini degli attributi in PK siano compatibili con i domini degli attributi in B

Vincoli globali generali (regole di business)

Sistema bancario con relazione CONTOCORRENTE e relazione MUTUO

- Esempio di vincolo: tutti i titolari di mutuo devono essere titolari di conto corrente

Questi vincoli sono suggeriti dalle regole imposte dal sistema informativo e sono detti **regole di business**

Definizione di schema di una base dati

La definizione dello schema S di una base dati è completata dai vincoli globali

12. Stato o istanza di una base di dati

Dato uno schema di base dati $S=\{R_1, R_2, \dots, R_n\}$, lo **stato della base di dati** è definito come

$$DB = (\langle R_1, r_1 \rangle, \langle R_2, r_2 \rangle, \dots, \langle R_n, r_n \rangle)$$

dove ogni r_i deve essere corretta e devono essere soddisfatti tutti i vincoli globali

Considerazione finale sul modello relazionale

Il modello relazionale è un
modello di dati orientato ai valori

- Quando costruisco informazioni che coinvolgono tuple di relazioni diverse non posso che lavorare sui valori contenuti nelle tuple
- L'identificazione (chiave) di una relazione è basata sui valori presenti nella chiave

In altri modelli (gerarchico, reticolare), le corrispondenze tra dati avvengono attraverso riferimenti (puntatori)