



SAPIENZA
UNIVERSITÀ DI ROMA

Facoltà di Ing. dell'Informazione, Informatica e Statistica
Laurea in Informatica

Basi di Dati, Modulo 2

Prof. Toni Mancini, Prof. Federico Mari

Dipartimento di Informatica

<http://tmancini.di.uniroma1.it>

<http://mari.di.uniroma1.it>

Slides B.2.1 (S.B.2.1)

Basi di Dati Relazionali

Sistemi di Gestione di Basi di Dati Relazionali
Il Modello Relazionale dei Dati

Indice

Queste slide sono composte dalle seguenti sottounità:

S.B.2.1.1. Relazioni e Tabelle

S.B.2.1.2. Informazione Incompleta

S.B.2.1.3. Vincoli di Integrità

S.B.2.1.3.1. Introduzione

S.B.2.1.3.2. Vincoli Intra-Relazionali

S.B.2.1.3.3. Vincoli Inter-Relazionali



SAPIENZA
UNIVERSITÀ DI ROMA

Facoltà di Ing. dell'Informazione, Informatica e Statistica
Laurea in Informatica

Basi di Dati, Modulo 2

Prof. Toni Mancini, Prof. Federico Mari

Dipartimento di Informatica

<http://tmancini.di.uniroma1.it>

<http://mari.di.uniroma1.it>

Slides B.2.1.1 (S.B.2.1.1)

Basi di Dati Relazionali

Sistemi di Gestione di Basi di Dati Relazionali

Il Modello Relazionale dei Dati

Relazioni e Tabelle

Il Modello Relazionale dei Dati

Prima di procedere con la Fase di Progettazione, richiamiamo le principali caratteristiche della tecnologia delle **Basi di Dati** e dei **DataBase Management Systems (DBMS) relazionali**.

Questa tecnologia offre funzionalità di **memorizzazione persistente** basate sul cosiddetto

Modello Relazionale dei Dati.

Il modello relazionale

- ▶ Proposto da E.F. Codd nel 1970 per favorire l'**indipendenza dei dati** dalle strutture di memorizzazione fisica
- ▶ Disponibile come modello logico in DBMS reali **solo** nel 1981 (non è stato facile realizzare l'indipendenza con efficienza e affidabilità)
- ▶ Si basa sul concetto di **relazione matematica** (con una variante)
- ▶ Le relazioni hanno una **rappresentazione naturale** per mezzo di **tabelle**
- ▶ Il modello è **basato sui valori**: anche i **riferimenti** tra dati in strutture (relazioni) diverse sono rappresentati per mezzo dei valori stessi

Prodotto cartesiano

Il modello relazionale si basa sul concetto di **relazione matematica**, che a sua volta si basa su quello di **prodotto cartesiano**

Definizione

Il **prodotto cartesiano** tra n insiemi D_1, \dots, D_n è l'insieme formato da tutte le possibili ennuple del tipo $\langle d_1, \dots, d_n \rangle$ tali che, per ogni $1 \leq i \leq n$:

$$d_i \in D_i$$

Prodotto cartesiano (2)

Esempio. Siano:

$$\blacktriangleright D_1 = \{a, b, c\}$$

$$\blacktriangleright D_2 = \{X, Y, Z\}$$

Il prodotto cartesiano $D_1 \times D_2$ è l'insieme di tutte le **coppie** il cui primo elemento è un membro di D_1 ed il cui secondo elemento è un elemento di D_2 :

$$D_1 \times D_2 = \left\{ \begin{array}{l} \langle a, X \rangle \\ \langle a, Y \rangle \\ \langle a, Z \rangle \\ \langle b, X \rangle \\ \langle b, Y \rangle \\ \langle b, Z \rangle \\ \langle c, X \rangle \\ \langle c, Y \rangle \\ \langle c, Z \rangle \end{array} \right\}$$

Nota: i domini sono **ordinati**, ad es., la coppia $\langle X, a \rangle \notin D_1 \times D_2$

Relazione matematica

Definizione

Dati n insiemi D_1, \dots, D_n , una **relazione matematica** r su D_1, \dots, D_n è un **sottoinsieme del prodotto cartesiano** tra D_1, \dots, D_n :

$$r \subseteq D_1 \times \dots \times D_n$$

Una relazione matematica quindi contiene **solo** le ennuple di D_1, \dots, D_n i cui elementi sono effettivamente legati dalla relazione

- ▶ la relazione è un **insieme**, quindi:
 - ▶ **non c'è un ordinamento** tra le ennuple
 - ▶ le ennuple sono tutte **distinte**
- ▶ ciascuna ennupla è **ordinata**, in quanto l' i -esimo valore appartiene all' i -esimo dominio.

Relazione matematica (2)

Esempio. Siano:

- ▶ *Stringa* l'insieme di tutte le stringhe alfanumeriche
- ▶ *InteroNonNeg* l'insieme di tutti gli interi non negativi.

Il prodotto cartesiano

$$\textit{Stringa} \times \textit{Stringa} \times \textit{InteroNonNeg} \times \textit{InteroNonNeg}$$

è l'insieme di tutte le possibili quadruple i cui primi due elementi sono membri di *Stringa* e i cui secondi due elementi sono membri di *InteroNonNeg*.

Relazione matematica (3)

Una relazione matematica

$$Partita \subseteq Stringa \times Stringa \times InterNonNeg \times InterNonNeg$$

potrebbe essere la seguente:

Roma	Lazio	1	1
Juventus	Milan	0	1
Lazio	Roma	2	1
Milan	Lazio	2	2

che descrive l'esito di alcune partite di calcio.

In quanto è un insieme, *Partita* non può contenere ennuple uguali, e non esiste alcun ordinamento tra le ennuple.

Relazione matematica (4)

Roma	Lazio	1	1
Juventus	Milan	0	1
Lazio	Roma	2	1
Milan	Lazio	2	2

Ciascuno dei due domini ha due **ruoli** distinti, distinguibili dalla **posizione** (i valori in ogni ennupla sono ordinati):

- ▶ il primo e il terzo dominio si riferiscono al nome ed al numero di goal della **squadra di casa**
- ▶ il secondo e il quarto dominio si riferiscono al nome ed al numero di goal della **squadra ospite**.

⇒ struttura **posizionale**.

Relazioni nel modello relazionale dei dati

Il concetto di relazione matematica viene **esteso** associando a ciascun dominio un nome (**attributo**), **unico** nella relazione, che “descrive” il ruolo del dominio

Esempio:

squadra casa	squadra ospite	goal casa	goal ospite
Roma	Lazio	1	1
Juventus	Milan	0	1
Lazio	Roma	2	1
Milan	Lazio	2	2

L'ordinamento fra attributi è **irrilevante**.

⇒ struttura **non posizionale**

Notazione

- ▶ Una relazione con attributi A_1, \dots, A_n sui domini, rispettivamente D_1, \dots, D_n può essere denotata mediante:

$$R(A_1 : D_1, \dots, A_n : D_n)$$

- ▶ Se t è una ennupla di R e A è un attributo di R , allora $t[A]$ indica il valore di t su A . Ad esempio, per $t = \langle Roma, Lazio, 1, 1 \rangle \in \text{Partita}$:

$$t[\text{squadra casa}] = Roma$$

Notazione (2)

- La stessa notazione può essere estesa ad insiemi di attributi, nel qual caso denota ennuple. Ad esempio:

$$t[\text{squadra casa}, \text{goal casa}] = \langle \text{Roma}, 1 \rangle \text{ (ennupla su due attributi)}$$

- Usando una notazione alternativa, è possibile vedere le ennuple di R come **ennuple etichettate**, in cui ogni valore è etichettato con il nome dell'attributo al quale si riferisce. Ad esempio:

$$t = \langle \text{squadra casa} : \text{Roma}, \text{squadra ospite} : \text{Lazio}, \\ \text{goal casa} : 1, \text{goal ospite} : 1 \rangle$$

Relazioni e tabelle

- Una relazione può essere rappresentata in modo naturale mediante una tabella, ad esempio:

squadra casa	squadra ospite	goal casa	goal ospite
Roma	Lazio	1	1
Juventus	Milan	0	1
Lazio	Roma	2	1
Milan	Lazio	2	2

- ma **non** tutte le tabelle rappresentano relazioni!

Una tabella rappresenta una relazione se:

- le intestazioni delle colonne (**attributi**) sono tutte diverse
- i valori di ciascuna colonna (attributo) sono tutti sullo stesso **dominio**
- le **righe** sono tutte **diverse** tra loro

Relazioni e tabelle (2)

squadra casa	squadra ospite	goal casa	goal ospite
Roma	Lazio	1	1
Juventus	Milan	0	1
Lazio	Roma	2	1
Milan	Lazio	2	2

In una tabella che rappresenta una relazione:

- l'ordine delle righe è **irrilevante**
- l'ordine delle colonne è **irrilevante**

Riferimenti tra i dati

Nel modello relazionale, i **riferimenti** tra i dati di diverse relazioni sono rappresentati dai **valori** che compaiono nelle ennuple.

Esempio

Studente				Corso		
mat	cognome	nome	nascita	codice	nome	aula
1101	Rossi	Mario	3/2/1990	590	Basi di dati	1C
1102	Bianchi	Anna	12/4/1989	591	Ing. SW	3F
1103	Verdi	Giulia	15/7/1995	592	Algebra	1C
				593	Geometria	3F

Esame		
studente	corso	voto
1101	590	28
1101	591	30
1103	593	25

Riferimenti tra i dati (2)

Nel modello relazionale, i **riferimenti** tra i dati di diverse relazioni sono rappresentati dai **valori** che compaiono nelle ennuple

Diversi vantaggi rispetto a **puntatori**:

- ▶ indipendenza dalle strutture fisiche di rappresentazione, che possono cambiare anche dinamicamente
- ▶ si rappresenta solo ciò che è di interesse per l'utente (l'uso di puntatori espliciti sarebbe "artificioso" per l'utente finale)
- ▶ i dati sono pienamente **portabili** da un sistema all'altro
- ▶ i riferimenti mediante valori possono essere percorsi in **entrambe le direzioni** (puntatori espliciti sarebbero uni-direzionali)

I puntatori possono comunque essere usati a livello fisico.

Definizioni

- **Schema di relazione**: nome di relazione più lista di attributi con domini:

$$R(A_1 : D_1, \dots, A_n : D_n)$$

- **Schema di base di dati**: insieme di schemi di relazione con nomi diversi:

$$\{R_1(X_1), \dots, R_m(X_m)\}$$

dove X_1, \dots, X_m sono gli insiemi di attributi di, rispettivamente, R_1, \dots, R_m

- (Istanza di) **Relazione** su uno schema $R(X)$: insieme di ennuple su X
- (Istanza di) **Base di dati** su uno schema $\{R_1(X_1), \dots, R_m(X_m)\}$: insieme $\{r_1, \dots, r_m\}$ tali che, per ogni $1 \leq i \leq m$, r_i è una istanza di relazione sullo schema $R_i(X_i)$



SAPIENZA
UNIVERSITÀ DI ROMA

Facoltà di Ing. dell'Informazione, Informatica e Statistica
Laurea in Informatica

Basi di Dati, Modulo 2

Prof. Toni Mancini, Prof. Federico Mari

Dipartimento di Informatica

<http://tmancini.di.uniroma1.it>

<http://mari.di.uniroma1.it>

Slides B.2.1.2 (S.B.2.1.2)

Basi di Dati Relazionali

Sistemi di Gestione di Basi di Dati Relazionali

Il Modello Relazionale dei Dati

Informazione Incompleta

Informazione incompleta

Il modello relazionale impone ai dati una **struttura rigida**:

- ▶ i dati sono rappresentati per mezzo di ennuple
- ▶ le ennuple ammesse sono dettate dagli schemi di relazione

Alcuni dati potrebbero non essere disponibili, per varie ragioni:

nome	prefettura
Roma	via IV Novembre 119/A
Firenze	
Tivoli	
Ogliastro	

Informazione incompleta (2)

nome	prefettura
Roma	via IV Novembre 119/A
Firenze	
Tivoli	
Ogliastro	

- ▶ di Firenze non conosciamo l'indirizzo della prefettura
- ▶ Tivoli non ha la prefettura
- ▶ Ogliastro è una provincia di recente istituzione: ha una prefettura?

Informazione incompleta: soluzioni?

Una possibile soluzione per gestire informazione incompleta è quella di usare **valori ordinari** del dominio (ad es., stringa vuota, il numero 0, il numero -1, etc.)

Approccio pericoloso:

- ▶ Potrebbero non esistere valori del dominio che non sono mai utilizzati
- ▶ Valori attualmente inutilizzati potrebbero essere utilizzati in seguito
- ▶ Tutti i programmi che accedono ai dati devono conoscere e tener conto del significato di questi valori.

Informazione incompleta: valori nulli

Si adotta una tecnica rudimentale ma efficace:

- ▶ **valore nullo (NULL)**: valore speciale che denota **assenza di informazione**
- ▶ Viene **esteso il concetto di ennupla**: una ennupla t sugli attributi $A_1 : D_1, \dots, A_n : D_n$ è tale che, per ogni attributo A_i , $t[A_i]$ appartiene a D_i oppure è **NULL**
- ▶ È importante imporre **restrizioni** sulla presenza di valori **NULL**.

Informazione incompleta: valori nulli (2)

Un valore **NULL** può essere **interpretato** in almeno tre modi diversi:

1. **Valore sconosciuto**: esiste un valore del dominio, ma non è noto (ad es., l'indirizzo della prefettura di Firenze)
2. **Valore inesistente**: non esiste un valore del dominio (ad es., l'indirizzo della prefettura di Tivoli)
3. **Valore senza informazione**: non è noto se esista o meno un valore del dominio (ad es., l'indirizzo della prefettura di Ogliastro).

I DBMS **non distinguono** tra questi tre tipi di informazione (quindi adottano implicitamente l'interpretazione di **valore senza informazione**).



SAPIENZA
UNIVERSITÀ DI ROMA

Facoltà di Ing. dell'Informazione, Informatica e Statistica
Laurea in Informatica

Basi di Dati, Modulo 2

Prof. Toni Mancini, Prof. Federico Mari

Dipartimento di Informatica

<http://tmancini.di.uniroma1.it>

<http://mari.di.uniroma1.it>

Slides B.2.1.3 (S.B.2.1.3)

Basi di Dati Relazionali

Sistemi di Gestione di Basi di Dati Relazionali

Il Modello Relazionale dei Dati

Vincoli di Integrità



SAPIENZA
UNIVERSITÀ DI ROMA

Facoltà di Ing. dell'Informazione, Informatica e Statistica
Laurea in Informatica

Basi di Dati, Modulo 2

Prof. Toni Mancini, Prof. Federico Mari
Dipartimento di Informatica

<http://tmancini.di.uniroma1.it>

<http://mari.di.uniroma1.it>

Slides B.2.1.3.1 (S.B.2.1.3.1)

Basi di Dati Relazionali
Sistemi di Gestione di Basi di Dati Relazionali
Il Modello Relazionale dei Dati
Vincoli di Integrità
Introduzione

Vincoli di integrità

Esistono istanze di una base dati che, pur sintatticamente corrette, non rappresentano informazioni possibili per la realtà di interesse:

Studente				Corso		
mat	cognome	nome	nascita	codice	nome	aula
1101	Rossi	Mario	3/2/1990	590	Basi di dati	1C
1102	Bianchi	Anna	12/4/1989	591	Ing. SW	3F
1103	Verdi	Giulia	15/7/1995	592	NULL	1C
				593	Geometria	3F

Esame			
studente	corso	voto	lode
1101	590	35	false
3399	591	30	true
1103	593	25	true

Il corso 592 non ha nome; 3399 non è la matricola di uno studente; 35 non è un voto possibile; non è possibile concedere la lode se il voto è 25.

Vincoli di integrità

Definizione. Proprietà che **deve** essere soddisfatta dalle istanze che rappresentano informazioni corrette per l'applicazione.

Ad uno schema di base di dati associamo un insieme di vincoli di integrità e consideriamo **legali** solo le istanze che li soddisfano tutti.

Tipi di vincoli:

- ▶ **intra**-relazionali
- ▶ **inter**-relazionali.

Vincoli di integrità (2)

Motivazioni

- ▶ Permettono di descrivere la realtà di interesse in modo più accurato di quanto le strutture (relazioni) permettano
- ▶ Contribuiscono ad aumentare la **qualità dei dati**
- ▶ Aiutano nella progettazione
- ▶ Sono utilizzati dal DBMS per scegliere la **strategia** di esecuzione delle interrogazioni.

Nota: non tutti i vincoli di interesse sono rappresentabili come vincoli di integrità esprimibili direttamente.



SAPIENZA
UNIVERSITÀ DI ROMA

Facoltà di Ing. dell'Informazione, Informatica e Statistica
Laurea in Informatica

Basi di Dati, Modulo 2

Prof. Toni Mancini, Prof. Federico Mari
Dipartimento di Informatica

<http://tmancini.di.uniroma1.it>

<http://mari.di.uniroma1.it>

Slides B.2.1.3.2 (S.B.2.1.3.2)

Basi di Dati Relazionali
Sistemi di Gestione di Basi di Dati Relazionali
Il Modello Relazionale dei Dati
Vincoli di Integrità
Vincoli Intra-Relazionali

Vincoli intra-relazionali: vincoli di ennupla

Esprimono **condizioni sui valori di ciascuna ennupla**, indipendentemente dalle altre ennuple

Esempi (relazione **Esame**):

- ▶ $voto \geq 18 \wedge voto \leq 30$
- ▶ $lode = true \rightarrow voto = 30$

Un vincolo su un solo attributo è chiamato **vincolo di dominio**

Vincoli intra-relazionali: vincoli di chiave

Studente			
mat	cognome	nome	nascita
1101	Rossi	Mario	3/2/1990
1102	Bianchi	Anna	12/4/1989
1103	Verdi	Giulia	15/7/1995

- ▶ il numero di matricola identifica gli studenti: non esistono due ennuple in **Studente** con lo stesso valore per l'attributo **mat**
- ▶ i dati anagrafici identificano gli studenti: non esistono due ennuple in **Studente** con gli stessi valori per la terna di attributi **cognome**, **nome**, **nascita**
(è solo un esempio, questo vincolo non vale in generale)

Vincoli intra-relazionali: vincoli di chiave (2)

Definizione (super-chiave di una relazione). Insieme K di attributi per cui **non possono** esistere istanze della relazione con ennuple che coincidono nei valori di tutti gli attributi in K

Definizione (chiave di una relazione). Insieme K di attributi che sono una super-chiave della relazione e tale che, per ogni $k \in K$, l'insieme $K - \{k\}$ non è super-chiave.

Una chiave quindi è una super-chiave che **non** può essere **accorciata**

Equivalentemente, una chiave è una super-chiave **\subseteq -minimale**

Vincoli intra-relazionali: vincoli di chiave (3)

Esempio:

Studente

mat	cognome	nome	nascita
1101	Rossi	Mario	3/2/1990
1102	Bianchi	Anna	12/4/1989
1103	Verdi	Giulia	15/7/1995

- ▶ {mat} è una super-chiave ed una chiave
- ▶ {mat, cognome} è una super-chiave, ma non una chiave
- ▶ {mat, cognome, nome, nascita} è una super-chiave, ma non una chiave

Ogni relazione ha almeno una super-chiave e quindi almeno una chiave: l'insieme di tutti gli attributi (in quanto in una relazione non possono esistere due ennuple uguali)

Vincoli intra-relazionali: vincoli di chiave (4)

Esempio:

Studente			
mat	cognome	nome	nascita
1101	Rossi	Mario	3/2/1990
1102	Bianchi	Anna	12/4/1989
1103	Verdi	Giulia	15/7/1995

Gli attributi **nome** e **nascita** formano una chiave?

- ▶ non esistono due ennuple che coincidono nel valore di **nome** e **nascita**...
- ▶ ma questo è **sempre** vero?

Vincoli: schemi ed istanze

- ▶ I vincoli corrispondono a proprietà del mondo reale modellato dalla base dati
- ▶ I vincoli interessano a livello di **schema**: vogliamo che siano soddisfatti da **tutte e sole** le istanze legali della base dati
- ▶ Una particolare istanza può soddisfare altri vincoli (“per caso”)

Importanza delle chiavi

L'esistenza delle chiavi garantisce l'**accessibilità** di ciascun dato della base dati

Ogni **singolo valore** è **univocamente** accessibile tramite:

- ▶ il nome della relazione
- ▶ il valore della chiave
- ▶ il nome dell'attributo

Le chiavi sono lo strumento principale attraverso il quale vengono correlati i dati in relazioni diverse ("il modello relazionale è basato sui valori")

Chiavi e valori NULL

In presenza di valori **NULL**, i valori degli attributi che formano una chiave di una relazione:

- ▶ non permettono di identificare univocamente le ennuple della relazione
- ▶ non permettono di realizzare facilmente i riferimenti con dati di altre relazioni

Chiavi e valori NULL (2)

Esempio:

Studente				Corso		
mat	cognome	nome	nascita	codice	nome	aula
NULL	Rossi	Mario	3/2/1990	590	Basi di dati	1C
1102	Bianchi	Anna	NULL	591	Ing. SW	3F
NULL	Verdi	Giulia	15/7/1995	NULL	Algebra	1C
				NULL	Geometria	3F

Esame			
studente	corso	voto	lode
1102	590	28	false
NULL	591	30	true
1102	NULL	25	false

Chiavi primarie

- ▶ La presenza di valori **NULL** nelle chiavi deve essere **limitata**
- ▶ Soluzione pratica: per ogni relazione, si sceglie una chiave, la **chiave primaria**
- ▶ Le ennuple della relazione **non** possono avere valori **NULL** per gli attributi della chiave primaria
- ▶ Notazione: gli attributi che compongono la chiave primaria vengono sottolineati

Esempio:

Studente

<u>mat</u>	cognome	nome	nascita
1101	Rossi	Mario	3/2/1990
1102	Bianchi	Anna	NULL
1103	Verdi	Giulia	15/7/1995
1104	Rossi	Mario	5/7/1991



SAPIENZA
UNIVERSITÀ DI ROMA

Facoltà di Ing. dell'Informazione, Informatica e Statistica
Laurea in Informatica

Basi di Dati, Modulo 2

Prof. Toni Mancini, Prof. Federico Mari
Dipartimento di Informatica

<http://tmancini.di.uniroma1.it>

<http://mari.di.uniroma1.it>

Slides B.2.1.3.3 (S.B.2.1.3.3)

Basi di Dati Relazionali
Sistemi di Gestione di Basi di Dati Relazionali
Il Modello Relazionale dei Dati
Vincoli di Integrità
Vincoli Inter-Relazionali

Vincoli inter-relazionali: integrità referenziale

Dati in relazioni diverse sono correlate attraverso **valori comuni**, in particolare, attraverso valori delle **chiavi** (di solito **primarie**)

Vincolo di integrità referenziale o di **foreign key** fra l'insieme X di attributi della relazione R e la relazione S :

ogni ennupla t di R deve avere valori per gli attributi X che compaiono come valori degli attributi **chiave primaria** di una ennupla di S

I vincoli di integrità referenziale giocano un ruolo fondamentale nel concetto di “modello basato sui valori”.

Vincoli inter-relazionali: integrità referenziale (2)

Esempio

Studente				Corso		
<u>mat</u>	cognome	nome	nascita	<u>codice</u>	nome	aula
1101	Rossi	Mario	3/2/1990	590	Basi di dati	1C
1102	Bianchi	Anna	12/4/1989	591	Ing. SW	3F
1103	Verdi	Giulia	15/7/1995	592	Algebra	1C
				593	Geometria	3F

Esame

<u>studente</u>	<u>corso</u>	voto
1101	590	28
1101	591	30
1103	593	25

$$\pi_{\text{studente}}(\text{Esame}) \subseteq \pi_{\text{matricola}}(\text{Studente})$$

$$\pi_{\text{Corso}}(\text{Esame}) \subseteq \pi_{\text{codice}}(\text{Corso})$$

Vincoli inter-relazionali: integrità referenziale (3)

Esempio

Officina	
<u>nome</u>	indirizzo
FixIt	via delle Spighe 4
CarFix	via delle Betulle 32
MotorGo	piazza Turing 1

Riparazione		
<u>officina</u>	<u>codice</u>	veicolo
FixIt	1	HK 243 BW
CarFix	1	AA 662 XQ
FixIt	2	HK 243 BW

$$\pi_{\text{officina}}(\text{Riparazione}) \subseteq \pi_{\text{nome}}(\text{Officina})$$

$$\pi_{\text{veicolo}}(\text{Riparazione}) \subseteq \pi_{\text{targa}}(\text{Veicolo})$$

Veicolo	
<u>targa</u>	tipo
HK 243 BW	auto
AA 662 XQ	auto
GF 211 HA	moto

RicambioRip		
<u>officina</u>	<u>rip</u>	<u>ricambio</u>
FixIt	1	A755
FixIt	1	A788
CarFix	1	A991
FixIt	2	B332

$$\pi_{\text{officina}}^{\text{rip}}(\text{RicambioRip}) \subseteq \pi_{\text{officina}}^{\text{codice}}(\text{Riparazione})$$

Vincoli inter-relazionali: integrità referenziale (4)

Riparazione

<u>officina</u>	<u>codice</u>	veicolo
FixIt	1	HK 243 BW
CarFix	1	AA 662 XQ
FixIt	2	HK 243 BW

$$\pi_{\text{officina}}(\text{Riparazione}) \subseteq \pi_{\text{nome}}(\text{Officina})$$

$$\pi_{\text{veicolo}}(\text{Riparazione}) \subseteq \pi_{\text{targa}}(\text{Veicolo})$$

RicambioRip

<u>officina</u>	<u>rip</u>	<u>ricambio</u>
FixIt	1	A755
FixIt	1	A788
CarFix	2	A991
FixIt	2	B332

$$\pi_{\text{officina}}^{\text{rip}}(\text{RicambioRip}) \not\subseteq$$

$$\pi_{\text{officina}}^{\text{codice}}(\text{Riparazione})$$

Integrità referenziale e valori NULL

In presenza di valori **NULL** i vincoli di integrità referenziale sono resi meno restrittivi:

Esempio:

Impiegato				Progetto	
<u>matr</u>	cognome	nome	progetto	<u>nome</u>	budget
1001	Rossi	Mario	IDEA	IDEA	5M
1002	Verdi	Anna	SMART	SMART	3M
1003	Rossi	Mario	QUICK	QUICK	5M
1004	Verdi	Carlo	NULL	PEEK	1M

$$\sigma_{\text{progetto} \neq \text{NULL}} (\pi_{\text{progetto}}(\text{Impiegato})) \subseteq \pi_{\text{nome}}(\text{Progetto})$$

Integrità referenziale: azioni compensative

Supponiamo che l'istanza di una base dati sia:

Impiegato				Progetto	
<u>matr</u>	cognome	nome	progetto	<u>nome</u>	budget
1001	Rossi	Mario	IDEA	IDEA	5M
1002	Verdi	Anna	SMART	SMART	3M
1003	Rossi	Mario	QUICK	QUICK	5M
1004	Verdi	Carlo	IDEA	PEEK	1M

$$\sigma_{\text{progetto} \neq \text{NULL}}(\pi_{\text{progetto}}(\text{Impiegato})) \subseteq \pi_{\text{nome}}(\text{Progetto})$$

Come dovrebbe comportarsi un DBMS se si tentasse di **cancellare** l'ennupla **⟨QUICK, 5M⟩** dalla relazione **Progetto**?

Integrità referenziale: azioni compensative

Supponiamo che l'istanza di una base dati sia:

Impiegato				Progetto	
matr	cognome	nome	progetto	nome	budget
1001	Rossi	Mario	IDEA	IDEA	5M
1002	Verdi	Anna	SMART	SMART	3M
1003	Rossi	Mario	QUICK	QUICK	5M
1004	Verdi	Carlo	IDEA	PEEK	1M

$$\sigma_{\text{progetto} \neq \text{NULL}}(\pi_{\text{progetto}}(\text{Impiegato})) \subseteq \pi_{\text{nome}}(\text{Progetto})$$

Come dovrebbe comportarsi un DBMS se si tentasse di **cancellare** l'ennupla **<QUICK, 5M>** dalla relazione **Progetto**?

\implies il DBMS dovrebbe **rifiutare** l'operazione, in quanto, dopo la cancellazione, l'ennupla **<1003, Rossi, Mario, QUICK>** **violerebbe** il vincolo di integrità.

Questo è il **comportamento di default**.

Integrità referenziale: altre azioni compensative

I DBMS prevedono anche **altri** meccanismi per **reagire** a cambiamenti della base dati che violano vincoli di integrità

Questi meccanismi vengono chiamati **azioni compensative**

Introduzione di valori NULL

Impiegato				Progetto	
<u>matr</u>	cognome	nome	progetto	<u>nome</u>	budget
1001	Rossi	Mario	IDEA	IDEA	5M
1002	Verdi	Anna	SMART	SMART	3M
1003	Rossi	Mario	QUICK	QUICK	5M
1004	Verdi	Carlo	IDEA	PEEK	1M

$$\sigma_{\text{progetto} \neq \text{NULL}} (\pi_{\text{progetto}}(\text{Impiegato})) \subseteq \pi_{\text{nome}}(\text{Progetto})$$

Integrità referenziale: altre azioni compensative

I DBMS prevedono anche **altri** meccanismi per **reagire** a cambiamenti della base dati che violano vincoli di integrità

Questi meccanismi vengono chiamati **azioni compensative**

Introduzione di valori NULL

1. L'utente cancella **QUICK**, 5M da **Progetto**

Impiegato				Progetto	
<u>matr</u>	cognome	nome	progetto	<u>nome</u>	budget
1001	Rossi	Mario	IDEA	IDEA	5M
1002	Verdi	Anna	SMART	SMART	3M
1003	Rossi	Mario	QUICK	PEEK	1M
1004	Verdi	Carlo	IDEA		

Il vincolo di integrità referenziale è **violato**:

$$\sigma_{\text{progetto} \neq \text{NULL}} (\pi_{\text{progetto}}(\text{Impiegato})) \not\subseteq \pi_{\text{nome}}(\text{Progetto})$$

Integrità referenziale: altre azioni compensative

I DBMS prevedono anche **altri** meccanismi per **reagire** a cambiamenti della base dati che violano vincoli di integrità

Questi meccanismi vengono chiamati **azioni compensative**

Introduzione di valori NULL

2. Il DBMS sostituisce con **NULL** ogni occorrenza di **QUICK** da **Impiegato**

Impiegato				Progetto	
<u>matr</u>	cognome	nome	progetto	<u>nome</u>	budget
1001	Rossi	Mario	IDEA	IDEA	5M
1002	Verdi	Anna	SMART	SMART	3M
1003	Rossi	Mario	NULL	PEEK	1M
1004	Verdi	Carlo	IDEA		

Il vincolo di integrità referenziale è **di nuovo soddisfatto**:

$$\sigma_{progetto \neq NULL}(\pi_{progetto}(Impiegato)) \subseteq \pi_{nome}(Progetto)$$

Integrità referenziale: altre azioni compensative

I DBMS prevedono anche **altri** meccanismi per **reagire** a cambiamenti della base dati che violano vincoli di integrità

Questi meccanismi vengono chiamati **azioni compensative**

Eliminazione in cascata

Impiegato				Progetto	
<u>matr</u>	cognome	nome	progetto	<u>nome</u>	budget
1001	Rossi	Mario	IDEA	IDEA	5M
1002	Verdi	Anna	SMART	SMART	3M
1003	Rossi	Mario	QUICK	QUICK	5M
1004	Verdi	Carlo	IDEA	PEEK	1M

$$\sigma_{\text{progetto} \neq \text{NULL}} (\pi_{\text{progetto}}(\text{Impiegato})) \subseteq \pi_{\text{nome}}(\text{Progetto})$$

Integrità referenziale: altre azioni compensative

I DBMS prevedono anche **altri** meccanismi per **reagire** a cambiamenti della base dati che violano vincoli di integrità

Questi meccanismi vengono chiamati **azioni compensative**

Eliminazione in cascata

1. L'utente cancella **QUICK**, 5M da **Progetto**

Impiegato				Progetto	
<u>matr</u>	cognome	nome	progetto	<u>nome</u>	budget
1001	Rossi	Mario	IDEA	IDEA	5M
1002	Verdi	Anna	SMART	SMART	3M
1003	Rossi	Mario	QUICK	PEEK	1M
1004	Verdi	Carlo	IDEA		

Il vincolo di integrità referenziale è **violato**:

$$\sigma_{progetto \neq NULL}(\pi_{progetto}(Impiegato)) \not\subseteq \pi_{nome}(Progetto)$$

Integrità referenziale: altre azioni compensative

I DBMS prevedono anche **altri** meccanismi per **reagire** a cambiamenti della base dati che violano vincoli di integrità

Questi meccanismi vengono chiamati **azioni compensative**

Eliminazione in cascata

2. Il DBMS cancella ogni ennupla di **Impiegato** in cui occorre il progetto
QUICK

Impiegato				Progetto	
<u>matr</u>	cognome	nome	progetto	<u>nome</u>	budget
1001	Rossi	Mario	IDEA	IDEA	5M
1002	Verdi	Anna	SMART	SMART	3M
1004	Verdi	Carlo	IDEA	PEEK	1M

Il vincolo di integrità referenziale è **di nuovo soddisfatto**:

$$\sigma_{\text{progetto} \neq \text{NULL}} (\pi_{\text{progetto}}(\text{Impiegato})) \subseteq \pi_{\text{nome}}(\text{Progetto})$$

Azioni compensative: sommario

I DBMS prevedono tre diverse azioni compensative per reagire ad aggiornamenti che violerebbero i vincoli di integrità referenziale:

- ▶ rifiuto dell'operazione (comportamento di default)
- ▶ introduzione di valori NULL
- ▶ cancellazione in cascata.

Sta al progettista della base dati definire quale comportamento il DBMS deve seguire per ogni singola relazione.