

Basi di Dati

prof. Letizia Tanca

Il Modello Relazionale dei Dati

lucidi tratti dal libro:

Atzeni, Ceri, Paraboschi, Torlone

“Introduzione alle Basi di dati”, Mc Graw
Hill Italia

I modelli logici dei dati

- Tradizionalmente, esistono tre modelli logici:
 - **gerarchico**
 - **reticolare**
 - **relazionale**
 - ultimi 10 anni: **a oggetti**
- I modelli gerarchico e reticolare sono più vicini alle strutture fisiche di memorizzazione, mentre il modello relazionale è più astratto:
 - nel modello relazionale si rappresentano solo valori — anche i riferimenti fra dati in strutture (relazioni) diverse sono rappresentati per mezzo dei valori stessi;
 - nei modelli gerarchico e reticolare si utilizzano riferimenti espliciti (puntatori) fra record.
- il modello **a oggetti** è un ibrido tra queste due concezioni, poiché è di livello alto ma fa uso di riferimenti

Il modello relazionale

- Proposto da E. F. Codd nel 1970 per favorire l'indipendenza dei dati e reso disponibile come modello logico in DBMS reali nel 1981 (non è facile implementare l'indipendenza con efficienza e affidabilità!).
- Si basa sul concetto matematico di **relazione** (con una variante).
- Le relazioni hanno una rappresentazione naturale per mezzo di tabelle.

Relazione: tre accezioni

- **relazione matematica**: come nella teoria degli insiemi;
- relazione (dall'inglese **relationship**) che rappresenta una classe di fatti — una relazione matematica fra due **entità**, nel modello **Entity-Relationship**; talvolta tradotto con **associazione** o **correlazione**
- **relazione** secondo il modello relazionale dei dati.

Relazione matematica

- D_1, D_2, \dots, D_n
- **prodotto cartesiano** $D_1 \times D_2 \times \dots \times D_n$ (n-uple ordinate (d_1, d_2, \dots, d_n) tali che $d_1 \in D_1, d_2 \in D_2, \dots, d_n \in D_n$)
- R relazione matematica su D_1, D_2, \dots, D_n è
$$R \subseteq D_1 \times D_2 \times \dots \times D_n.$$
- D_1, D_2, \dots, D_n **domini** della relazione R . n **grado** di R .
- il numero di n-uple è la **cardinalità** della relazione. Nelle applicazioni reali, la cardinalità è sempre **finita**.

Relazione matematica, proprietà

- una relazione è un **insieme**; quindi:
 - non è definito alcun ordinamento fra le n-uple;
 - le n-uple di una relazione sono distinte l'una dall'altra;
- le n-uple sono **ordinate**: l' i-esimo valore di ciascuna proviene dall' i -esimo dominio; è cioè definito un ordinamento fra i domini.

Relazione matematica, esempio

Partite \subseteq *string* \times *string* \times *integer* \times *integer*

| | | | |
|-------|-------|---|---|
| Juve | Lazio | 3 | 1 |
| Lazio | Milan | 2 | 0 |
| Juve | Roma | 1 | 2 |
| Roma | Milan | 0 | 1 |

- La struttura è **posizionale**

Relazioni nel modello relazionale dei dati

- A ciascun dominio associamo un nome (**attributo**)
- L'ordinamento fra gli attributi è irrilevante:
la struttura è **non posizionale**

| Casa | Fuori | RetiCasa | RetiFuori |
|-------|-------|----------|-----------|
| Juve | Lazio | 3 | 1 |
| Lazio | Milan | 2 | 0 |
| Juve | Roma | 1 | 2 |
| Roma | Milan | 0 | 1 |

Formalizzando

- L'associazione fra domini e attributi è definita da una funzione $dom: X \rightarrow D$ che associa a ciascun attributo un dominio.
- Una **ennupla** (o **tupla**) su un insieme di attributi X è una funzione che associa a ciascun attributo A in X un valore del dominio $dom(A)$
- Una **relazione** su X è un insieme di tuple su X

Notazioni

- Se t è una tupla su X e $A \in X$, allora $t[A]$ (o $t.A$) indica il valore di t su A .
- la prima si chiama notazione funzionale, la seconda notazione punto (o dot notation)
- Nell'esempio, se t è la prima tupla della tabella,

$$t[\textit{Fuori}] = \textit{Lazio}$$

Tabelle e relazioni

- Una tabella rappresenta una relazione se
 - i valori di ciascuna **colonna** sono fra loro **omogenei** (dallo stesso dominio)
 - le **righe** sono **diverse** fra loro
 - le **intestazioni delle colonne** sono **diverse** tra loro (perciò i campi sono distinguibili **mediante il loro nome e non la loro posizione**)
- Inoltre, in una tabella che rappresenta una relazione
 - **l'ordinamento tra le righe è irrilevante**
 - **l'ordinamento tra le colonne è irrilevante**

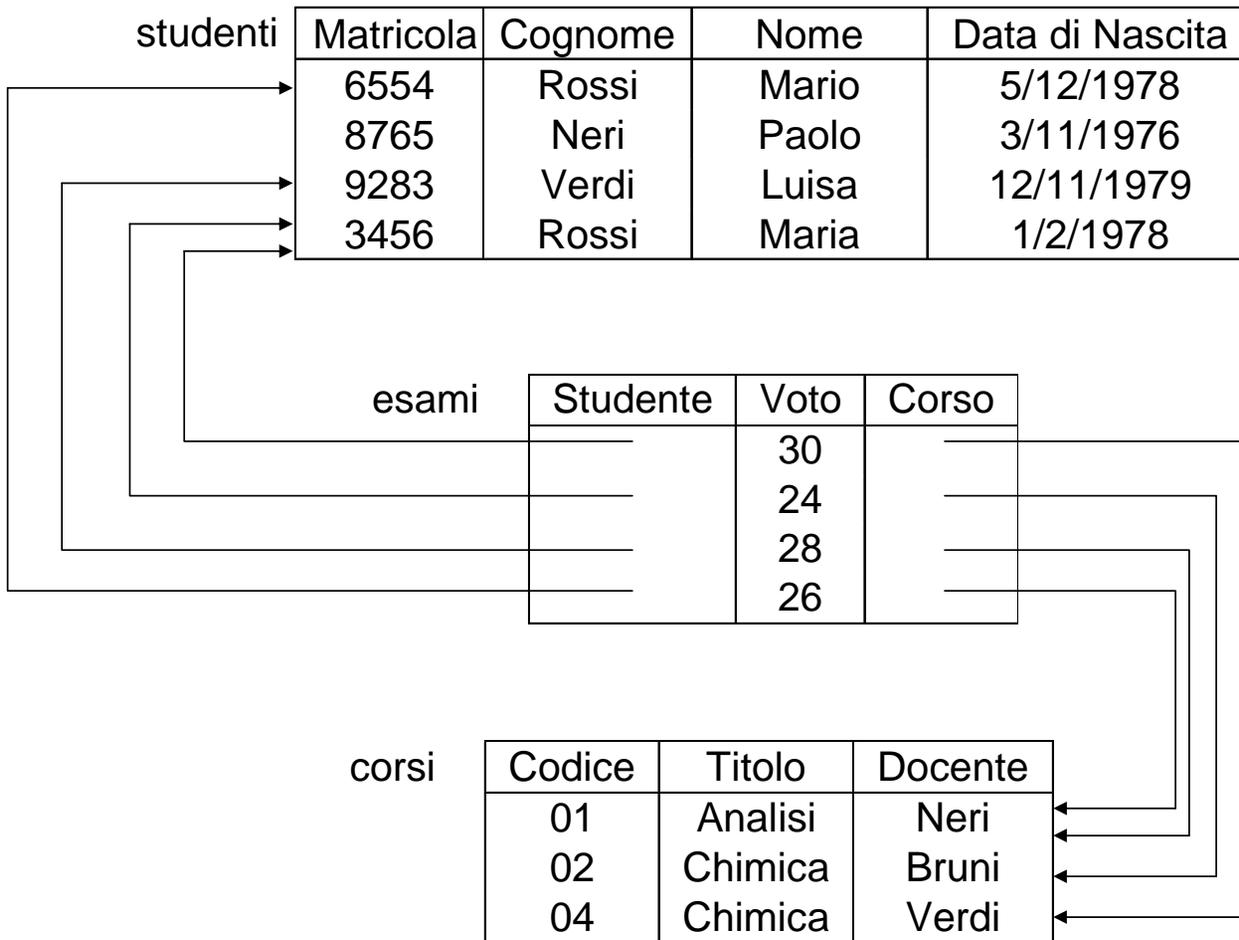
Il modello relazionale è basato su valori

I **referimenti fra dati** in relazioni diverse sono rappresentati **per mezzo di valori** dei domini che compaiono nelle tuple.

| studenti | Matricola | Cognome | Nome | Data di Nascita |
|----------|-----------|---------|-------|-----------------|
| | 6554 | Rossi | Mario | 5/12/1978 |
| | 8765 | Neri | Paolo | 3/11/1976 |
| | 9283 | Verdi | Luisa | 12/11/1979 |
| | 3456 | Rossi | Maria | 1/2/1978 |

| esami | Studente | Voto | Corso |
|-------|----------|------|-------|
| | 3456 | 30 | 04 |
| | 3456 | 24 | 02 |
| | 9283 | 28 | 01 |
| | 6554 | 26 | 01 |

| corsi | Codice | Titolo | Docente |
|-------|--------|---------|---------|
| | 01 | Analisi | Neri |
| | 02 | Chimica | Bruni |
| | 04 | Chimica | Verdi |



Vantaggi della struttura basata su valori

- **indipendenza** dalle strutture fisiche (si potrebbe avere anche con puntatori di alto livello) che possono cambiare anche dinamicamente
- si rappresenta **solo ciò che è rilevante** dal punto di vista dell'applicazione (dell'utente); i puntatori sono meno comprensibili per l'utente finale (senza, l'utente finale vede gli stessi dati dei programmatori)
- i dati sono **portabili piu' facilmente** da un sistema ad un altro
- i puntatori **sono direzionali**

NOTE:

- i puntatori **possono esistere a livello fisico**
- nel modello a oggetti esistono i riferimenti, che funzionano da “puntatori di alto livello „

Definizioni

Schema di relazione: un **nome di relazione** R con un insieme di **attributi** A_1, \dots, A_n

$$R(A_1, \dots, A_n)$$

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

$$R = \{R_1(X_1), \dots, R_n(X_n)\}$$

(Istanza di) relazione su uno schema $R(X)$: insieme r di tuple su X

(Istanza di) base di dati su uno schema

$$R = \{R_1(X_1), \dots, R_n(X_n)\}:$$

insieme di relazioni $r = \{r_1, \dots, r_n\}$ (con r_i relazione su R_i)

Terminologia

attributi

| Corsi | Codice | Nome | Docente |
|--------------|--------|---------------------------|---------|
| tuple | M2170 | Fondamenti di informatica | Verdi |
| | M4880 | Sistemi di Elaborazione | Bianchi |
| | F0410 | Basi di dati | Neri |



dominio: insieme dei codici
dei corsi tenuti al Politecnico

Esempio

- sono possibili relazioni su un solo attributo

| studenti | Matricola | Cognome | Nome | Data di Nascita |
|----------|-----------|---------|-------|-----------------|
| | 6554 | Rossi | Mario | 5/12/1978 |
| | 8765 | Neri | Paolo | 3/11/1976 |
| | 9283 | Verdi | Luisa | 12/11/1979 |
| | 3456 | Rossi | Maria | 1/2/1978 |

| Studenti lavoratori | Matricola |
|---------------------|-----------|
| | 6554 |
| | 8765 |

Strutture nidificate

| "Da Filippo" Via Roma 23, Chissadove | | |
|--|-----------|-------|
| Ricevuta 2369 del 12/05/1997 | | |
| 3 | Coperti | 6000 |
| 2 | Antipasti | 12000 |
| 3 | Primi | 27000 |
| 2 | Bistecche | 36000 |
| Totale | | 81000 |

| "Da Filippo" Via Roma 23, Chissadove | | |
|--|-----------|-------|
| Ricevuta 2456 del 16/05/1997 | | |
| 2 | Coperti | 4000 |
| 1 | Antipasti | 6000 |
| 2 | Primi | 15000 |
| 2 | Orate | 50000 |
| 2 | Caffè | 3000 |
| Totale | | 78000 |

Basi di Dati

19

Rappresentazione di strutture nidificate per mezzo di relazioni

ricevute

| Numero | Data | Totale |
|--------|------------|--------|
| 2369 | 12/05/1997 | 81000 |
| 2456 | 16/05/1997 | 78000 |

dettaglio

| Numero | Quantità | Descrizione | Importo |
|--------|----------|-------------|---------|
| 2369 | 3 | Coperti | 6000 |
| 2369 | 2 | Antipasti | 12000 |
| 2369 | 3 | Primi | 27000 |
| 2369 | 2 | Bistecche | 36000 |
| 2456 | 2 | Coperti | 4000 |
| 2456 | 1 | Antipasti | 6000 |
| 2456 | 2 | Primi | 15000 |
| 2456 | 2 | Orate | 50000 |
| 2456 | 2 | Caffè | 3000 |

**Queste relazioni sono in prima forma normale:
ogni tupla contiene solo elementi atomici**

Informazione incompleta

- Il modello relazionale impone ai dati una struttura rigida:
 - le informazioni sono rappresentate per mezzo di tuple
 - solo alcuni formati di tuple sono ammessi: quelli che corrispondono agli schemi di relazione
- I dati disponibili possono non corrispondere esattamente al formato previsto, per varie ragioni.

Informazione incompleta: motivazioni

- **Firenze è provincia, ma non conosciamo l'indirizzo della prefettura**
- **Tivoli non è provincia: non ha prefettura**
- **Prato è “nuova” provincia: ha la prefettura?**

| Città | Prefettura |
|---------|-----------------|
| Roma | Via IV novembre |
| Firenze | |
| Tivoli | |
| Prato | |

Informazione incompleta: soluzioni?

- non conviene (anche se spesso si fa) utilizzare valori ordinari del dominio (0, stringa nulla, “99”, etc), per vari motivi:
 - potrebbero non esistere valori “non utilizzati”
 - valori “non utilizzati” potrebbero diventare significativi
 - in fase di utilizzo (ad esempio, nei programmi) sarebbe necessario ogni volta tener conto del “significato” di questi valori

Informazione incompleta nel modello relazionale

- Si adotta una tecnica rudimentale ma efficace:
 - **valore nullo**: denota l'assenza di un valore del dominio (e non è un valore del dominio)
- Formalmente, è sufficiente estendere il concetto di tupla: $t[A]$, per ogni attributo A , è un valore del dominio $\text{dom}(A)$ oppure il valore nullo NULL
- Si possono (e debbono) imporre restrizioni sulla presenza di valori nulli

Troppi valori nulli!

| Studenti | Matricola | Cognome | Nome | Nascita |
|----------|-----------|---------|-------|------------|
| | 276545 | Rossi | Maria | NULL |
| | NULL | Neri | Anna | 23/04/1972 |
| | NULL | Verdi | Fabio | 12/02/1972 |

| Esami | Studente | Voto | Corso |
|-------|----------|------|-------|
| | 276545 | 28 | 01 |
| | NULL | 27 | NULL |
| | 200768 | 24 | NULL |

| Corsi | Codice | Titolo | Docente |
|-------|--------|---------|---------|
| | 01 | Analisi | Giani |
| | 03 | NULL | NULL |
| | NULL | Chimica | Belli |

Tipi di valore nullo

- (almeno) tre casi differenti
 - **valore sconosciuto:** esiste un valore del dominio, ma non è noto (Firenze)
 - **valore inesistente:** non esiste un valore del dominio (Tivoli)
 - **valore senza informazione:** non è noto se esista o meno un valore del dominio (Prato)
- I DBMS non distinguono i tipi di valore nullo (e quindi implicitamente adottano il valore **senza informazione**)

Vincoli di integrità

- Esistono istanze di basi di dati che, pur sintatticamente corrette, **non rappresentano informazioni possibili** (cioè sensate) per l'applicazione di interesse.

| Studenti | Matricola | Cognome | Nome | Nascita |
|----------|-----------|---------|-------|------------|
| | 276545 | Rossi | Maria | 23/04/1968 |
| | 276545 | Neri | Anna | 23/04/1972 |
| | 788854 | Verdi | Fabio | 12/02/1972 |

| Esami | Studente | Voto | Lode | Corso |
|-------|----------|------|--------|-------|
| | 276545 | 28 | e lode | 01 |
| | 276545 | 32 | | 02 |
| | 788854 | 23 | | 03 |
| | 200768 | 30 | e lode | 03 |

| Corsi | Codice | Titolo | Docente |
|-------|--------|---------|---------|
| | 01 | Analisi | Giani |
| | 03 | NULL | NULL |
| | 02 | Chimica | Belli |

Vincolo di integrità

- Definizione
 - proprietà che **deve essere soddisfatta dalle istanze che rappresentano informazioni corrette**
 - ogni vincolo può essere visto come una funzione booleana (o un predicato) che associa ad ogni istanza il valore **vero** o **falso**.
- Tipi di vincoli:
 - vincoli intrarelazionali; casi particolari:
 - vincoli su valori (o di dominio)
 - vincoli di tupla
 - vincoli interrelazionali

Vincoli di integrità, motivazioni

- **risultano utili al fine di descrivere la realtà di interesse in modo più accurato di quanto le strutture permettano;**
- **forniscono un contributo verso la “qualità dei dati”**
- **costituiscono uno strumento di ausilio alla progettazione**
- **possono essere utilizzati dal sistema nella scelta della strategia di esecuzione delle interrogazioni**
- **molti di essi si esprimono mediante il linguaggio associato al DBMS prescelto**
- **purtroppo non tutte le proprietà di interesse sono rappresentabili per mezzo di vincoli esprimibili direttamente nel linguaggio**

Vincoli di tupla

- Esprimono condizioni sui valori di ciascuna tupla, **indipendentemente dalle altre tuple.**
- Una possibile sintassi: **espressione booleana** (con AND, OR e NOT) di atomi che confrontano valori di attributo o espressioni aritmetiche su di essi.
- Un vincolo di tupla è un **vincolo di dominio** se **coinvolge un solo attributo**
- Esempi (in una notazione logica):

(VOTO ≥18) AND (VOTO ≤30)

(VOTO =30) OR NOT (LODE = "E LODE")

LORDO = (RITENUTE + NETTO)

Identificazione delle tuple

| Matricola | Cognome | Nome | Corso | Nascita |
|-----------|---------|-------|-------------|-----------|
| 6554 | Rossi | Mario | Informatica | 5/12/1978 |
| 8765 | Rossi | Mario | Informatica | 3/11/1976 |
| 4723 | Verdi | Laura | Meccanica | 10/7/1979 |
| 9283 | Verdi | Mario | Informatica | 3/11/1976 |
| 3456 | Rossi | Laura | Meccanica | 5/12/1978 |

- **il numero di matricola identifica gli studenti:**
 - **non ci sono due tuple con lo stesso valore sull'attributo Matricola**
- **i dati anagrafici identificano gli studenti:**
 - **non ci sono due tuple uguali su tutti e tre gli attributi Cognome, Nome e Data di Nascita**

Vincoli di chiave

- **chiave:**
insieme di attributi che **identificano univocamente** le tuple di una relazione
- più precisamente:
 - un insieme K di attributi è **superchiave** per una relazione r se r non contiene due tuple distinte t_1 e t_2 con $t_1[K] = t_2[K]$
 - K è **chiave** per r se è una superchiave minimale (cioè non contiene un'altra superchiave) per r

| Matricola | Cognome | Nome | Corso | Nascita |
|-----------|---------|-------|-------------|-----------|
| 6554 | Rossi | Mario | Informatica | 5/12/1978 |
| 8765 | Rossi | Mario | Informatica | 3/11/1976 |
| 4723 | Verdi | Laura | Meccanica | 10/7/1979 |
| 9283 | Verdi | Mario | Informatica | 3/11/1976 |
| 3456 | Rossi | Laura | Meccanica | 5/12/1978 |

- **Matricola è una chiave:**
 - **Matricola è superchiave**
 - **contiene un solo attributo e quindi è minimale**
- **Cognome, Nome, Nascita è un'altra chiave:**
 - **l'insieme Cognome, Nome, Nascita è superchiave**
 - **nessuno dei suoi sottoinsiemi è superchiave (minimalità)**

| Matricola | Cognome | Nome | Corso | Nascita |
|-----------|---------|-------|-------------|-----------|
| 6554 | Rossi | Mario | Informatica | 5/12/1978 |
| 8765 | Rossi | Mario | Elettronica | 3/11/1976 |
| 4723 | Verdi | Laura | Meccanica | 10/7/1979 |
| 9283 | Verdi | Mario | Informatica | 3/11/1976 |
| 3456 | Rossi | Laura | Meccanica | 5/12/1978 |

NOTA BENE:

- la relazione **non contiene tuple fra loro uguali su Cognome e Corso:**
 - in ogni corso di studio gli studenti hanno cognomi diversi;
 - l'insieme { Cognome, Corso } è superchiave minimale e quindi chiave
- **possiamo dire che questa proprietà è sempre soddisfatta?**
 - No! In generale ci possono essere in un corso di studio studenti con lo stesso cognome

Chiavi, schemi e istanze

- i vincoli **corrispondono a proprietà del mondo reale** modellato dalla base di dati;
- quindi **interessano a livello di schema** (con riferimento cioè a tutte le istanze):
 - ad uno schema associamo un insieme di vincoli e consideriamo **corrette** (lecite, valide, ammissibili) solo le istanze che soddisfano tutti i vincoli;
 - singole istanze possono soddisfare ulteriori vincoli (“per pura coincidenza”)

Individuazione delle chiavi

- definendo uno schema di relazione, associamo ad esso **i vincoli di chiave** che vogliamo siano soddisfatti dalle sue istanze (corrette)
- li individuiamo
 - considerando **le proprietà che i dati soddisfano nell'applicazione** (il “frammento di mondo reale di interesse”);
 - notando quali insiemi di attributi **permettono di identificare univocamente le tuple**;
 - e individuando **i sottoinsiemi minimali** di tali insiemi che conservano la capacità di identificare le tuple.

Individuazione delle chiavi, esempio

- Allo schema di relazione
STUDENTI(Matricola, Cognome, Nome, Corso, Nascita)
associamo i vincoli che indicano come chiavi gli insiemi di attributi Matricola e Cognome, Nome, Nascita
- La relazione

| Matricola | Cognome | Nome | Corso | Nascita |
|-----------|---------|-------|-------------|-----------|
| 6554 | Rossi | Mario | Informatica | 5/12/1978 |
| 8765 | Rossi | Mario | Elettronica | 3/11/1976 |
| 4723 | Verdi | Laura | Meccanica | 10/7/1979 |
| 9283 | Verdi | Mario | Informatica | 3/11/1976 |
| 3456 | Rossi | Laura | Meccanica | 5/12/1978 |

è corretta, perché soddisfa i vincoli associati allo schema.

- Ne soddisfa anche altri. Ad esempio, Cognome, Corso è chiave per essa.

Teorema di esistenza delle chiavi

**Ogni schema di relazione ha (almeno)
una chiave**

- poiché **le relazioni sono insiemi**, ogni relazione non può contenere tuple distinte ma uguali fra loro:
ogni relazione ha come superchiave l'insieme degli attributi su cui è definita;
- perciò ogni schema di relazione ha almeno una chiave (l'insieme di tutti gli attributi o un suo sottoinsieme).

Importanza delle chiavi

- l'esistenza delle chiavi **garantisce l'accessibilità a ciascun dato della base di dati**
- ogni singolo valore è univocamente accessibile tramite:
 - nome della relazione
 - valore della chiave
 - nome dell'attributo(i)
- le chiavi sono **lo strumento principale attraverso il quale vengono correlati i dati** in relazioni diverse (“il modello relazionale è basato su valori”)

Chiavi e valori nulli

- In presenza di valori nulli, i valori degli attributi che formano la chiave
 - non permettono di identificare le tuple come desiderato
 - né permettono di realizzare facilmente i riferimenti da altre relazioni

| Matricola | Cognome | Nome | Nascita | Corso |
|-----------|---------|-------|----------|-------------|
| NULL | Rossi | Luca | NULL | Informatica |
| 8765 | Rossi | Mario | 01/05/61 | Civile |
| 4856 | Neri | Mario | NULL | NULL |
| NULL | Neri | Mario | 05/03/63 | Civile |

Chiave primaria

- La presenza di valori nulli nelle chiavi deve essere limitata
- Soluzione pratica: per ogni relazione scegliamo una chiave (la chiave primaria) su cui **non ammettiamo valori nulli**.
- Notazione per la chiave primaria: gli attributi che la compongono sono sottolineati

| <u>Matricola</u> | Cognome | Nome | Nascita | Corso |
|------------------|---------|-------|----------|-------------|
| 6554 | Rossi | Luca | NULL | Informatica |
| 8765 | Rossi | Mario | 01/05/61 | Civile |
| 4856 | Neri | Mario | NULL | NULL |
| 6590 | Neri | Mario | 05/03/63 | Civile |

Vincoli di integrità referenziale ("foreign key")

- informazioni in relazioni diverse sono correlate **attraverso valori comuni**
- in particolare, valori delle **chiavi** (primarie, di solito)
- un **vincolo di integrità referenziale** fra un insieme di attributi X di una relazione R_1 e un'altra relazione R_2 impone ai valori su X di ciascuna tupla dell'istanza di R_1 di comparire come valori della chiave (primaria) dell'istanza di R_2

Base di dati con vincoli di integrità referenziale

infrazioni

| Codice | Data | Vigile | Prov | Numero |
|--------|-----------|--------|------|--------|
| 65524 | 3/9/1997 | 343 | MI | 3K9886 |
| 87635 | 4/12/1997 | 476 | MI | 6D5563 |
| 82236 | 4/12/1997 | 343 | RM | 7C5567 |
| 35632 | 6/1/1998 | 476 | RM | 7C5567 |
| 76543 | 5/3/1998 | 548 | MI | 6D5563 |

vigili

| <u>Matricola</u> | Cognome | Nome |
|------------------|----------|------|
| 343 | Rossi | Luca |
| 476 | Neri | Pino |
| 548 | Nicolosi | Gino |

automobili

| <u>Prov</u> | <u>Numero</u> | Proprietario | ... |
|-------------|---------------|--------------|-----|
| MI | 3K9886 | Nestore | ... |
| MI | 6D5563 | Nestore | ... |
| RM | 7C5567 | Menconi | ... |
| RM | 1A6673 | Mussone | ... |
| MI | 5E7653 | Marchi | ... |

Base di dati con vincoli di integrità referenziale

Nell'esempio, esistono vincoli di integrità referenziale fra:

- l'attributo Vigile della relazione INFRAZIONI e la relazione VIGILI
- gli attributi Prov e Numero di INFRAZIONI e la relazione AUTO

Base di dati che viola vincoli di integrità referenziale

infrazioni

| Codice | Data | Vigile | Prov | Numero |
|--------|-----------|--------|------|--------|
| 65524 | 3/9/1997 | 343 | MI | 3K9886 |
| 87635 | 4/12/1997 | 476 | MI | 6D5563 |
| 82236 | 4/12/1997 | 343 | RM | 7C5567 |
| 35632 | 6/1/1998 | 476 | RM | 7C5567 |
| 76543 | 5/3/1998 | 548 | MI | 6D5563 |

vigili

| <u>Matricola</u> | Cognome | Nome |
|------------------|----------|------|
| 343 | Rossi | Luca |
| 548 | Nicolosi | Gino |

automobili

| <u>Prov</u> | <u>Numero</u> | Proprietario | ... |
|-------------|---------------|--------------|-----|
| MI | 3K9886 | Nestore | ... |
| RM | 6D5563 | Nestore | ... |
| MI | 7C5567 | Menconi | ... |
| RM | 1A6673 | Mussone | ... |
| MI | 5E7653 | Marchi | ... |

Vincoli di integrità referenziale: commenti

- I vincoli di integrità referenziale **giocano un ruolo fondamentale** nel concetto “modello relazionale basato su valori.”
- Sono possibili **meccanismi per il supporto** alla gestione dei vincoli di integrità referenziale (“azioni” da svolgere in corrispondenza a violazioni).
- Se si ammettono valori nulli i vincoli possono essere resi meno restrittivi
- Attenzione ai **vincoli su più attributi**

Base di dati con vincoli di integrità referenziale

incidenti

| <u>Codice</u> | Data | ProvA | NumeroA | ProvB | NumeroB |
|---------------|-----------|-------|---------|-------|---------|
| 65524 | 3/9/1997 | MI | 3K9886 | MI | 7C5567 |
| 87635 | 4/12/1997 | RM | 6D5563 | RM | 1A6673 |
| 82236 | 6/12/1997 | MI | 7C5567 | RM | 6D5563 |

automobili

| <u>Prov</u> | <u>Numero</u> | Proprietario | ... |
|-------------|---------------|--------------|-----|
| MI | 3K9886 | Nestore | ... |
| RM | 6D5563 | Nestore | ... |
| MI | 7C5567 | Menconi | ... |
| RM | 1A6673 | Mussone | ... |
| MI | 5E7653 | Marchi | ... |

Le dipendenze funzionali

- I vincoli di chiave sono dei particolari tipi di vincoli, parte di una categoria di vincoli più vasta: le dipendenze funzionali
 - Dati due insiemi di attributi X e Y
 - Si dice che X determina Y, o che Y dipende funzionalmente da X, e si scrive $X \rightarrow Y$ sse:
***date due tuple distinte t_1 e t_2 ,
se $t_1[X] = t_2[X]$ allora $t_1[Y] = t_2[Y]$***
- Nella parte di progettazione studieremo come si usano le dipendenze funzionali per garantire opportune proprietà della BD