

Basi di dati e Relazioni

Studenti (`Matricola`, `Cognome`, `Nome`, `DataNascita`)

Corsi (`Codice`, `Titolo`, `Docente`)

Esami (`Studente`, `Voto`, `Corso`)

Studenti contiene dati su un insieme di studenti

Corsi contiene dati su un insieme di corsi

Esami contiene dati su un insieme di esami e **fa riferimento** alle altre due attraverso i numeri di matricola e il codice del corso.

Quindi `Matricola` e `Studente`, come `Corso` e `Codice`, sono definiti sullo stesso dominio e possono (in questo caso devono, per generare il riferimento) assumere gli stessi valori.

studenti

Matricola	Cognome	Nome	Data di nascita
6554	Rossi	Mario	05/12/1978
8765	Neri	Paolo	03/11/1976
9283	Verdi	Luisa	12/11/1979
3456	Rossi	Maria	01/02/1978

esami

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

corsi

Codice	Titolo	Docente
01	Analisi	Mario
02	Chimica	Bruni
04	Chimica	Verdi

studenti

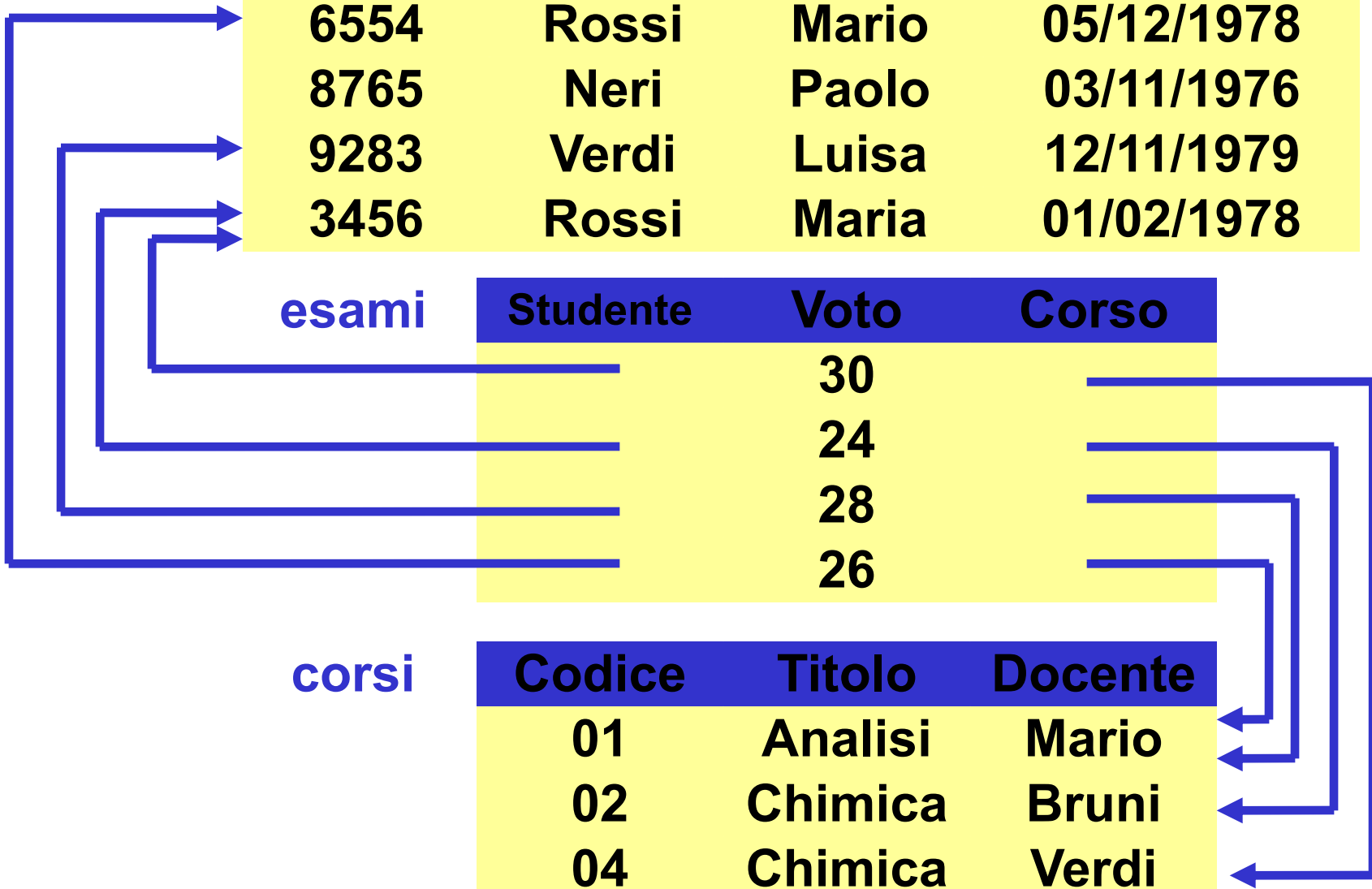
Matricola	Cognome	Nome	Data di nascita
6554	Rossi	Mario	05/12/1978
8765	Neri	Paolo	03/11/1976
9283	Verdi	Luisa	12/11/1979
3456	Rossi	Maria	01/02/1978

esami

Studente	Voto	Corso
	30	
	24	
	28	
	26	

corsi

Codice	Titolo	Docente
01	Analisi	Mario
02	Chimica	Bruni
04	Chimica	Verdi



Basi di dati e Relazioni

Dall'esempio abbiamo visto che:

- Il modello relazionale è *basato su valori*.
- I riferimenti fra dati in relazioni diverse avvengono attraverso la *corrispondenza dei valori* con i quali attributi corrispondenti vengono istanziati nelle tuple che sono logicamente correlate.

NB *Attributi corrispondenti* implica una corrispondenza diretta sul nome oppure (logica) sul dominio, specificata attraverso l'imposizione di opportuni vincoli (in genere per i due attributi sono ammissibili gli stessi valori o, per uno dei due, sono ammissibili i valori di un sottoinsieme dei valori con cui è stato istanziato l'altro).

Gli altri modelli (gerarchico, reticolare) utilizzano *puntatori* per realizzare la corrispondenza.

Basi di dati e Relazioni

Vantaggi dell'approccio basato su valori

- Si inseriscono nella base di dati solo valori significativi per l'applicazione (i puntatori sono dati aggiuntivi relativi alla sola implementazione fisica).
- Il trasferimento dei dati da un ambiente ad un altro è più semplice (i puntatori hanno validità solo locale)
- la rappresentazione logica dei dati non fa riferimento a quella fisica e quindi si ottiene l'indipendenza (fisica) dei dati

Informazione incompleta

Le tuple che compongono la base di dati devono essere omogenee. Quindi, *in **ogni** tupla, ad **ogni** attributo* deve essere associato un valore. Non sempre questo è possibile.

Es. Persone(Cognome, Nome, Indirizzo, Telefono)

Potrebbe esistere una persona che non possiede telefono, o di cui non conosciamo l'indirizzo.

Cognome	Nome	Indirizzo	Telefono
Rossi	Francesco	v. Rossa 22	0521 335643
Verdi	Giuseppe	v. Verde 11	
Bruni	Bruno		336 7564213

Questa relazione non è ammissibile! In ogni tupla, ogni attributo deve essere istanziato.

Informazione incompleta

- non conviene (anche se è un espediente di uso comune) usare valori del dominio normalmente non utilizzati (0, stringa nulla, “99”, ...), come spesso accade nella programmazione:
 - potrebbero non esistere valori “non utilizzati”
 - valori “non utilizzati” potrebbero, a un certo punto, diventare significativi
 - in fase di utilizzo (nei programmi) sarebbe necessario ogni volta tener conto del “significato” (non standardizzato) di questi valori

Informazione incompleta

Nel modello relazionale è definito un valore convenzionale, detto *valore nullo*, che indica la *non disponibilità* dell'informazione.

Il valore nullo può rappresentare correttamente tre situazioni logicamente diverse in cui l'informazione è:

- Sconosciuta (*so che il valore esiste ma non lo conosco*)
- Inesistente (*so che il valore non esiste*)
- indeterminata (*non so se il valore esiste e, in ogni caso, non lo conosco*)

Vincoli di Integrità

Non tutte le combinazioni possibili di valori dei domini su cui è definita una relazione sono accettabili.

- Alcuni attributi possono assumere valori in un certo intervallo
- Alcuni attributi  **devono**  essere diversi in ogni tupla della stessa relazione

Es. valori dell'attributo Matricola nella relazione

Studenti (`Matricola`, `Cognome`, `Nome`, `DataNascita`)

Vincoli di Integrità

- Alcuni valori possono essere incompatibili con altri all'interno della stessa relazione

Es. data la relazione

`Esami(Matricola, Voto, Lode, CodCorso)`

1. una stessa coppia Matricola,Corso può apparire una sola volta
2. Il valore Vero per l'attributo Lode è corretto solo se Voto=30

Vincoli di Integrità

- Alcuni valori possono essere incompatibili con i valori di un'altra relazione

Es. date le relazioni

`Studenti (Matricola, Cognome, Nome, DataNascita)`

`Esami (Studente, Voto, Lode, CodCorso)`

`Corsi (CodCorso, Titolo, Docente)`

ogni valore di `CodCorso` in `Esami`, in questo caso, **DEVE** essere un valore esistente di `CodCorso` nella relazione `Corsi`;

analogamente, ogni valore dell'attributo `Studente` nella tabella `Esami` **DEVE** essere un valore esistente dell'attributo `Matricola` nella relazione `Studenti`.

Vincoli di Integrità

Sono condizioni, espresse come *predicati logici*, inserite nella base di dati per garantirne la consistenza.

Ogni istanza della base di dati deve soddisfare i vincoli di integrità, cioè

il predicato corrispondente al vincolo deve assumere valore vero in ogni istante.

Una istanza che soddisfi tutti i vincoli è detta *corretta* (o *lecita*, o *ammissibile*)

Vincoli di Integrità

Un vincolo è detto:

- **Intrarelazionale**

se coinvolge attributi della stessa relazione

Es.

- **Vincoli di tupla**

possono essere valutati su ciascuna tupla
indipendentemente dalle altre

- **Vincoli di dominio**

definiti su singoli valori

- **Interrelazionale** se coinvolge più relazioni

Vincoli di Tupla

Possono essere definiti attraverso operatori booleani

Es. Data la relazione

`Esami (Matricola, Voto, Lode, CodCorso)`

`(Voto >= 18) AND (Voto <= 30)`

`(NOT (lode=Vero)) OR (Voto=30)`

oppure, data la relazione

`Pagamenti (Data, Importo, Ritenute, Netto)`

`Netto = Importo - Ritenute`

Identificazione delle tuple

Matricola	Cognome	Nome	Corso	Nascita
27655	Rossi	Mario	Ing Inf	5/12/78
78763	Rossi	Mario	Ing Inf	3/11/76
65432	Neri	Piero	Ing Mecc	10/7/79
87654	Neri	Mario	Ing Inf	3/11/76
67653	Rossi	Piero	Ing Mecc	5/12/78

Osservazioni:

- non ci sono due tuple con lo stesso valore sull'attributo Matricola
- non ci sono due tuple uguali su tutti e tre gli attributi Cognome, Nome e Nascita

Chiavi

Una *chiave* è un insieme *minimale* di attributi utilizzato per **identificare univocamente** le tuple di una relazione.

Formalmente:

Un insieme di attributi K è *superchiave* per una relazione r se r non contiene due tuple t_1 e t_2 tali che

$$t_1[K] = t_2[K]$$

Un **insieme di attributi** K è **chiave** per r **se è superchiave minimale**, cioè se non esiste un'altra superchiave K' che sia sottoinsieme di K o, comunque, di grado inferiore.

Esempio

Matricola	Cognome	Nome	Corso	Nascita
27655	Rossi	Mario	Ing Inf	5/12/78
78763	Rossi	Mario	Ing Inf	3/11/76
65432	Neri	Piero	Ing Mecc	10/7/79
87654	Neri	Mario	Ing Inf	3/11/76
67653	Rossi	Piero	Ing Mecc	5/12/78

- **Matricola è una chiave:**
 - è superchiave
 - contiene un solo attributo e quindi è minimale

Chiavi

Una chiave è tale se soddisfa la definizione per tutte le possibili tuple appartenenti alla relazione, e non solo per quelle che effettivamente appaiono come istanze della relazione stessa.

==> la chiave è *legata allo schema* della relazione e non ai valori effettivamente assunti dalle istanze dello schema.

Ogni relazione, per definizione, possiede una chiave.

Infatti, poiché una relazione non ammette due tuple uguali, l'intero insieme di attributi X su cui è definita è sicuramente superchiave per la relazione.

Chiave primaria

La presenza di valori nulli in una chiave può vanificare la proprietà di unicità delle tuple che identifica.

Si impone quindi che almeno una chiave non contenga valori nulli. Tale chiave è detta *chiave primaria*.

Di solito la chiave primaria compare sottolineata nello schema di una relazione.

Es.

Studenti (Matricola, Cognome, Nome, Nascita, Corso)