

Lezione 6 – Progettazione di una base di dati relazionale – Problemi, e vincoli

Prof.ssa Maria De Marsico
demarsico@di.uniroma1.it



SAPIENZA
UNIVERSITÀ DI ROMA

- Supponiamo di voler creare una base di dati
- contenente i seguenti dati di studenti universitari:
 - dati anagrafici e identificativi
 - nome e cognome,
 - data, comune e provincia di nascita, matricola,
 - codice fiscale
 - dati curriculari
 - per ogni esame sostenuto:
 - voto,
 - data,
 - codice, titolo e docente del corso

-
- La base di dati consiste di una sola relazione con schema:
- Curriculum (Matr, CF, Cogn, Nome, DataN, Com, Prov, C#, Tit, Doc, DataE, Voto)

Ipotesi 1 (problemi)



SAPIENZA
UNIVERSITÀ DI ROMA
DIPARTIMENTO DI INFORMATICA

Curriculum	Matr	CF	Cogn	Nome	DataN	ComN	ProvN	C#	TitC	DocC	DataE	Voto
	01	...	Rossi	Mario	...	Tolfa	Roma	10	Fisica	Pippo
	02	...	Bianchi	Paolo	...	Tolfa	Roma	10	Fisica	Pippo
	01	...	Rossi	Mario	...	Tolfa	Roma	20	Chimica	Pluto

Ipotesi 1 (problemi)



SAPIENZA
UNIVERSITÀ DI ROMA
DIPARTIMENTO DI INFORMATICA

Curriculum	Matr	CF	Cogn	Nome	DataN	ComN	ProvN	C#	TitC	DocC	DataE	Voto
	01	...	Rossi	Mario	...	Tolfa	Roma	10	Fisica	Pippo
	02	...	Bianchi	Paolo	...	Tolfa	Roma	10	Fisica	Pippo
	01	...	Rossi	Mario	...	Tolfa	Roma	20	Chimica	Pluto

Ridondanza

- I dati anagrafici di uno studente sono memorizzati per ogni esame sostenuto dallo studente
- I dati di un corso sono memorizzati per ogni esame sostenuto per quel corso



- La ridondanza da luogo a:
 - spreco di spazio in memoria
 - **anomalie** di aggiornamento, inserimento e cancellazione

Ipotesi 1 (problemi)



Curriculum	Matr	C F	Cogn	Nome	DataN	ComN	ProvN	C#	TitC	DocC	DataE	Voto
	01	...	Rossi	Mario	...	Tolfa	Roma	10	Fisica	Pippo
	02	...	Bianchi	Paolo	...	Tolfa	Roma	10	Fisica	Pippo
	01	...	Rossi	Mario	...	Tolfa	Roma	20	Chimica	Pluto

Anomalia di aggiornamento

- se cambia il docente del corso il dato deve essere modificato per **ogni esame** sostenuto per quel corso

Curriculum	Matr	C F	Cogn	Nome	DataN	ComN	ProvN	C#	TitC	DocC	DataE	Voto
	01	...	Rossi	Mario	...	Tolfa	Roma	10	Fisica	Minni
	02	...	Bianchi	Paolo	...	Tolfa	Roma	10	Fisica	Minni
	01	...	Rossi	Mario	...	Tolfa	Roma	20	Chimica	Pluto

Ipotesi 1 (problemi)



Curriculum	Matr	CF	Cogn	Nome	DataN	ComN	ProvN	C#	TitC	DocC	DataE	Voto
	01	...	Rossi	Mario	...	Tolfa	Roma	10	Fisica	Pippo
	02	...	Bianchi	Paolo	...	Tolfa	Roma	10	Fisica	Pippo
	01	...	Rossi	Mario	...	Tolfa	Roma	20	Chimica	Pluto

Anomalia di inserimento

- non posso inserire i dati anagrafici di uno studente
finchè non ha sostenuto almeno un esame a meno
di non usare valori nulli (spreco di spazio!); idem per i corsi

Curriculum	Matr	CF	Cogn	Nome	DataN	ComN	ProvN	C#	TitC	DocC	DataE	Voto
	01	...	Rossi	Mario	...	Tolfa	Roma	10	Fisica	Minni
	02	...	Bianchi	Paolo	...	Tolfa	Roma	10	Fisica	Minni
	01	...	Rossi	Mario	...	Tolfa	Roma	20	Chimica	Pluto
	03	...	Neri	Giulio	...	Nepi	Roma	-	-	-	-	-

Ipotesi 1 (problemi)



Curriculum	Matr	CF	Cogn	Nome	DataN	ComN	ProvN	C#	TitC	DocC	DataE	Voto
	01	...	Rossi	Mario	...	Tolfa	Roma	10	Fisica	Pippo
	02	...	Bianchi	Paolo	...	Tolfa	Roma	10	Fisica	Pippo
	01	...	Rossi	Mario	...	Tolfa	Roma	20	Chimica	Pluto

Anomalia di cancellazione

- eliminando i dati anagrafici di uno studente
potrebbero essere eliminati i dati di un corso
(se lo studente è l'unico ad aver sostenuto l'esame
di quel corso); idem quando elimino un corso

Curriculum	Matr	CF	Cogn	Nome	DataN	ComN	ProvN	C#	TitC	DocC	DataE	Voto
	02	...	Bianchi	Paolo	...	Tolfa	Roma	10	Fisica	Pippo



La base di dati consiste di **tre** schemi di relazione:

- Studente (Matr, CF, Cogn, Nome, Data, Com, Prov)
- Corso (C#, Tit, Doc)
- Esame (Matr, C#, Data, Voto)

Ipotesi 2



I dati nella relazione:

Curriculum	Matr	CF	Cogn	Nome	DataN	Com	Prov	C#	Tit	Doc	DataE	Voto
	01	...	Rossi	Mario	...	Tolfa	Roma	10	Fisica	Pippo
	02	...	Bianchi	Paolo	...	Tolfa	Roma	10	Fisica	Pippo
	01	...	Rossi	Mario	...	Tolfa	Roma	20	Chimica	Pluto

possono essere memorizzati nelle tre relazioni

Studente	Matr	CF	Cogn	Nome	Data	Com	Prov	Corso	C#	Tit	Doc
	01	...	Rossi	Mario	...	Tolfa	Roma		10	Fisica	Pippo
	02	...	Bianchi	Paolo	...	Tolfa	Roma		20	Chimica	Pluto

Esame	Matr	C#	Data	Voto
	01	10
	02	10
	01	20



NON abbiamo più i problemi di ridondanza e le anomalie di aggiornamento, inserimento e cancellazione esaminati in precedenza però ...

Ipotesi 2



SAPIENZA
UNIVERSITÀ DI ROMA
DIPARTIMENTO DI INFORMATICA

Studente	Matr	CF	Cogn	Nome	Data	Com	Prov
	01	...	Rossi	Mario	...	Tolfa	Roma
	02	...	Bianchi	Paolo	...	Tolfa	Roma

Ridondanza

- Il fatto che un comune si trova in una certa provincia è ripetuto per ogni studente nato in quel comune

Anomalia di aggiornamento

- Se un comune cambia provincia (in seguito alla creazione di una nuova Provincia) devono essere modificate più tuple

Anomalia di inserimento

- Non è possibile memorizzare il fatto che un certo comune si trova in una certa provincia se non c'è almeno uno studente nato in quel comune

Anomalia di cancellazione

- Se vengono eliminati i dati anagrafici di uno studente potrebbe perdersi l'informazione che un certo comune si trova in una certa provincia (se è l'unico studente nato in quel comune)

La base di dati consiste di quattro schemi di relazione:

- Studente (Matr, CF, Cogn, Nome, Data, Com)
- Corso (C#, Tit, Doc)
- Esame (Matr, C#, Data, Voto)
- Comune (Com, Prov)

Ipotesi 3



SAPIENZA
UNIVERSITÀ DI ROMA
DIPARTIMENTO DI INFORMATICA

Studente	Matr	CF	Cogn	Nome	Data	Com
	01	...	Rossi	Mario	...	Tolfa
	02	...	Bianchi	Paolo	...	Tolfa

Comune	Com	Prov
	Tolfa	Roma

Esame	Matr	C#	Data	Voto
	01	10
	02	10
	01	20

Corso	C#	Tit	Doc
	10	Fisica	Pippo
	20	Chimica	Pluto

NON ho più ridondanza né anomalie!



Uno schema di basi di dati è “buono” se non presenta

- ridondanze
- anomalie di aggiornamento, inserimento e cancellazione

Quindi ...



- Studente (Matr, CF, Cogn, Nome, Data, Com)
- Corso (C#, Tit, Doc)
- Esame (Matr, C#, Data, Voto)
- Comune (Com, Prov)

... è uno schema “buono”



- Come progettare uno schema “buono”?

I problemi esaminati in relazione allo schema

Curriculum (Matr, CF, Cogn, Nome, DataN, Com, Prov, C#, Tit, Doc, DataE, Voto)

derivano dal fatto che sono rappresentati in **un'unica relazione** tre **concetti distinti** (studente, corso, esame) e ...



... vengono superati quando i tre concetti vengono rappresentati nelle tre relazioni distinte

- Studente (Matr, CF, Cogn, Nome, Data, Com, Prov)
- Corso (C#, Tit, Doc)
- Esame (Matr, C#, Data, Voto)

Analogamente i problemi esaminati in relazione allo schema

Studente (Matr, CF, Cogn, Nome, Data, Com, Prov)

derivano dal fatto che sono rappresentati in **un'unica relazione due concetti distinti** (studente, comune) e ...

... vengono superati quando i due concetti vengono rappresentati nelle due relazioni distinte

- Studente (Matr, CF, Cogn, Nome, Data, Com)
- Comune (Com, Prov)

Ovviamente possiamo (e dobbiamo) saltare immediatamente alla soluzione corretta ... nello schema iniziale i concetti diversi sono quattro, ma uno è meno evidente



- Spesso la cattiva progettazione, cioè in particolare l'errore di rappresentare più concetti nella stessa relazione, deriva dalla necessità di recuperare informazioni relative **non solo agli oggetti ma anche alle loro associazioni**.
- Es. nel caso appena visto, operazioni eseguite frequentemente potrebbero richiedere di recuperare i dati relativi **a tutti gli esami** sostenuti da uno studente o ai docenti dei corsi che ha frequentato, quindi potrebbe apparire **“comodo”** memorizzare tutte queste informazioni insieme.
- Abbiamo visto che le operazioni dell'algebra relazionale (e quelle dei linguaggi delle basi di dati relazionali) consentono attraverso i **referimenti per valore** tra gli oggetti associati e attraverso opportuni **join** tra relazioni di **ottenere le stesse informazioni senza incorrere nei problemi che abbiamo discusso**.



Per progettare uno schema “buono” occorre rappresentare **separatamente ogni concetto** in una relazione distinta



- Così abbiamo tutte le informazioni disponibili in un solo colpo Ma abbiamo tutte le anomalie di cui abbiamo parlato!

Ordini	Nome	C#	Città	A#	N-pezzi	Denom.	Prezzo
	Rossi	C1	Roma	A1	100	Piatto	3
	Rossi	C2	Milano	A2	200	Bicchiere	2
	Bianchi	C3	Roma	A2	150	Bicchiere	2
	Verdi	C4	Roma	A3	200	Tazza	4
	Rossi	C1	Roma	A2	200	Bicchiere	2
	Rossi	C1	Roma	A3	100	Tazza	4



Ordini	Nome	C#	Città	A#	N-pezzi	Denom.	Prezzo
	Rossi	C1	Roma	A1	100	Piatto	3
	Rossi	C2	Milano	A2	200	Bicchiere	2
	Bianchi	C3	Roma	A2	150	Bicchiere	2
	Verdi	C4	Roma	A3	200	Tazza	4
	Rossi	C1	Roma	A2	200	Bicchiere	2
	Rossi	C1	Roma	A3	100	Tazza	4

- Se devo aggiornare la città di un cliente, devo aggiornare TUTTE le tuple, e in caso di inconsistenze non sono più in grado di sapere quale è quella giusta!
- Se cancello un articolo che va fuori produzione, rischio di perdere i dati di un cliente che ha ordinato solo quell'articolo
- Per inserire un articolo deve esserci per forza un cliente che lo ordina

Una soluzione peggiore del male



SAPIENZA
UNIVERSITÀ DI ROMA
DIPARTIMENTO DI INFORMATICA

Ordini	Nome	C#	Città	A#	N-pezzi	Denom.	Prezzo
	Rossi	C1	Roma	A1	100	Piatto	3
	Rossi	C2	Milano	A2	200	Bicchiere	2
	Bianchi	C3	Roma	A2	150	Bicchiere	2
	Verdi	C4	Roma	A3	200	Tazza	4
	Rossi	C1	Roma	A2	200	Bicchiere	2
	Rossi	C1	Roma	A3	100	Tazza	4

Articolo

A#	Denom.	Prezzo
A1	Piatto	3
A2	Bicchiere	2
A3	Tazza	4

- Aumentiamo solo le probabilità di errore!



- Così abbiamo una base di dati “buona” e le informazioni possiamo comunque ricavarle tramite operazioni di join.

Cliente	Nome	C#	Città
	Rossi	C1	Roma
	Rossi	C2	Milano
	Bianchi	C3	Roma
	Verdi	C4	Roma

Articolo	A#	Denom.	Prezzo
	A1	Piatto	3
	A2	Bicchiere	2
	A3	Tazza	4

Ordine	C#	A#	N-pezzi
	C1	A1	100
	C2	A2	200
	C3	A2	150
	C4	A3	200
	C1	A2	200
	C1	A3	100



Come possono essere individuati i concetti rappresentati in una relazione (ben progettata)?

Possiamo individuare i concetti rappresentati in una relazione tramite una (la?)

chiave

che è un attributo o un gruppo di attributi
che determinano una particolare

dipendenza funzionale

che è un particolare tipo di **vincolo**



VINCOLI



Nella realtà che si vuole rappresentare in una base di dati sono soddisfatte certe **condizioni**. Ad esempio:

1. Un voto è un intero compreso tra 18 e 30
2. Il numero di matricola identifica univocamente uno studente
3. Il numero di matricola in un verbale di esame deve essere il numero di matricola di uno studente
4. Lo stipendio di un impiegato non può diminuire
5. Lo straordinario è dato dal numero di ore fatte per la paga oraria



- Quando si rappresenta una realtà di interesse in una base di dati deve essere possibile rappresentare anche tali condizioni.
- Un **vincolo** è la rappresentazione nello schema di una base di dati di una condizione valida nella realtà di interesse.
- Un'istanza della base di dati è **legale** se soddisfa tutti i vincoli (cioè se è una rappresentazione fedele della realtà).

- Un DBMS permette di
- **definire** insieme allo schema della base di dati i vincoli
- **verificare** che un'istanza della base di dati sia legale
- in base a speciali vincoli predefiniti, **impedire** l'inserimento di tuple che violerebbero tali vincoli

Definizione e verifica dei vincoli nei DBMS



SAPIENZA
UNIVERSITÀ DI ROMA
DIPARTIMENTO DI INFORMATICA

- Un DBMS è dotato di procedure per la verifica dei vincoli che ricorrono più frequentemente:
 - **vincoli di dominio** (un voto è un intero compreso tra 18 e 30)
 - **chiavi** (il numero di matricola identifica univocamente uno studente)
 - **contenimento di domini** (il numero di matricola in un verbale di esame deve essere il numero di matricola di uno studente)
- Per la verifica di altri tipi di vincoli (es: vincoli dinamici, vincoli che coinvolgono valori di più attributi in un'espressione matematica) può essere necessario definire opportune procedure.



Vedremo che le **dipendenze funzionali** definite su uno schema di relazione esprimono particolari vincoli di dipendenza **tra sottoinsiemi di attributi dello schema** stesso, che devono essere soddisfatti **da ogni istanza** dello schema