

Basi di Dati - IV

Corso di Laurea in Informatica
Anno Accademico 2013/2014

Paolo Baldan

baldan@math.unipd.it

<http://www.math.unipd.it/~baldan>

Progettazione Logica

Progettazione Logica: Modello Relazionale

3

- Il **modello dei dati relazionale** (Edgar F. Codd, 1970)
 - **Trasformazione** dal modello **concettuale ad oggetti** al modello **logico relazionale**
 - **Algebra relazionale**
-

Il Modello Relazionale

● **Collezioni** come **relazioni (tabelle)**

Studenti	
Nome:	string
Cognome:	string>
Matricola:	string <<PK>
AnnoCorso:	int

Esami	
Codice:	string <<PK>
Candidato:	string <<FK(Studenti)>>
Materia:	string
Data:	date
Voto:	int
Lode:	bool

● **Associazioni** tramite **chiavi**

Studenti	
Nome:	string
Cognome:	string>
Matricola:	string <<PK>
AnnoCorso:	int

Candidato ←

Esami	
Codice:	string <<PK>
Candidato:	string <<FK(Studenti)>>
Materia:	string
Data:	date
Voto:	int
Lode:	bool

● I meccanismi per definire una base di dati con il modello relazionale sono l'**ennupla** e la **relazione**.

● Dal punto di vista matematico dati D_1, \dots, D_n (domini)

● **ennupla** $\langle d_1, \dots, d_n \rangle \in D_1 \times D_2 \times \dots \times D_n$

- $d_1 \in D_1, \dots, d_n \in D_n$

● **relazione** $R \subseteq D_1 \times D_2 \times \dots \times D_n$

● in Informatica si associa un'etichetta distinta a ciascun dominio D_1, \dots, D_n (record!)

● **Tipo ennupla T**: insieme finito di coppie (**Attributo**, **Tipo primitivo**):
 $\langle A_1: T_1, \dots, A_n: T_n \rangle$

● **Tipo ennupla T**: insieme finito di coppie (**Attributo**, **Tipo primitivo**):
 $\langle A_1: T_1, \dots, A_n: T_n \rangle$

● **Schema di relazione**

$R : \{ T \}$ (T tipo ennupla, {T} tipo relazione)

Modello Relazionale (cont.)

7

- **Tipo ennupla T**: insieme finito di coppie (Attributo, Tipo primitivo):
<A1: T1, ..., An: Tn>
- **Schema di relazione**
 $R : \{ T \}$ (T tipo ennupla, {T} tipo relazione)
- Spesso scriveremo R(T) invece di R:{T}.

Modello Relazionale (cont.)

7

- **Tipo ennupla T**: insieme finito di coppie (Attributo, Tipo primitivo):
<A1: T1, ..., An: Tn>
- **Schema di relazione**
 $R : \{ T \}$ (T tipo ennupla, {T} tipo relazione)
- Spesso scriveremo R(T) invece di R:{T}.
- **Istanza** di uno schema R:{T}: insieme finito di ennuple di tipo T.

Modello Relazionale (cont.)

7

- **Tipo ennupla T**: insieme finito di coppie (Attributo, Tipo primitivo):
<A1: T1, ..., An: Tn>
- **Schema di relazione**
 $R : \{ T \}$ (T tipo ennupla, {T} tipo relazione)
- Spesso scriveremo R(T) invece di R:{T}.
- **Istanza** di uno schema R:{T}: insieme finito di ennuple di tipo T.
- **Schema relazionale di una BD**:
 - insieme di schemi di relazione $R_i:\{T_i\}$;
 - vincoli di integrità

Esempio

8

- Studenti (Nome: string, Cognome: string, Matricola: string, Anno:int)

Nome	Cognome	Matricola	Anno
Paolo	Verdi	71523	2005
Anna	Rossi	76366	2006
Giorgio	Zeri	71347	2005

Studenti

- se non interessa evidenziare il tipo degli attributi scriviamo
Studenti(Nome, Cognome, Matricola, Anno)

Schema:

Studenti (Nome: string, Cognome: string, Matricola: string, Anno: int)

Esami (Codice: string, Materia: string, Candidato: string, Data: string, Voto: int, Lode:char)

Studenti

Nome	Cognome	Matricola	Anno
Paolo	Verdi	71523	2005
Anna	Rossi	76366	2006
Giorgio	Zeri	71347	2005

Esami

Codice	Materia	Candidato	Data	Voto	Lode
B112	BD	71523	08.07.06	27	N
F31	FIS	76366	08.07.07	26	N
B247	CN	71523	28.12.06	30	S

Considereremo

- **chiavi**
- **chiavi esterne**
- **valori non nulli**

Chiavi

● **Superchiave in R:** sottoinsieme X di attributi di R tale che il valore degli attributi in X determina univocamente una ennupla

● **Esempio:** in `Studenti(Nome, Cognome, Matricola, Univ, CF, Anno)`
`(Matricola, Univ)` e `(Cognome, Matricola, Univ)`, `(CF)` superchiavi

Chiavi

● **Superchiave in R:** sottoinsieme X di attributi di R tale che il valore degli attributi in X determina univocamente una ennupla

● **Esempio:** in `Studenti(Nome, Cognome, Matricola, Univ, CF, Anno)`
`(Matricola, Univ)` e `(Cognome, Matricola, Univ)`, `(CF)` superchiavi

● **Chiave:** superchiave **minimale**: gli attributi che appartengono ad una chiave sono detti **primi**

● **Esempio:** `(Matricola, Univ)`, `(CF)`

- **Superchiave in R**: sottoinsieme X di attributi di R tale che il valore degli attributi in X determina univocamente una ennupla
 - Esempio: in `Studenti(Nome, Cognome, Matricola, Univ, CF, Anno)`
(`Matricola, Univ`) e (`Cognome, Matricola, Univ`), (`CF`) superchiavi
- **Chiave**: superchiave **minimale**: gli attributi che appartengono ad una chiave sono detti **primi**
 - Esempio: (`Matricola, Univ`), (`CF`)
- **Chiave primaria**: una delle chiavi, in genere di lunghezza minima

- **Chiave esterna in R**
 - insieme di attributi $X = \{A1, \dots, An\}$ di R che riferisce la chiave primaria $Y = \{B1, \dots, Bn\}$ di S:
 - per ogni ennupla r in R esiste una ennupla s in S t.c.
 $r.X = s.Y$ (r "riferisce" s). [**integrità referenziale**]
- **Associazioni**
 - realizzate con il meccanismo di chiave

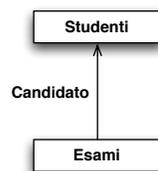
Esempio

Schema:

`Studenti(Nome: string, Cognome: string, Matricola: string, Anno: int)`

`Esami(Codice: string, Materia: string, Candidato*: string, Data: string, Voto: int, Lode: char)`

● Associazione:

**Studenti**

Nome	Cognome	<u>Matricola</u>	Anno
Paolo	Verdi	71523	2005
Anna	Rossi	76366	2006
Giorgio	Zeri	71347	2005

Esami

<u>Codice</u>	Materia	Candidato*	Data	Voto	Lode
B112	BD1	71523	08.07.06	27	N
F31	FIS	76366	08.07.07	26	N
B247	BD2	71523	28.12.06	30	S

Esempio: altre soluzioni

Esempio: altre soluzioni

14

- Studenti(Nome, Cognome, Matricola, Anno, Esame*)
Esami(Codice, Materia, Data, Voto, Lode)

Esempio: altre soluzioni

14

- Studenti(Nome, Cognome, Matricola, Anno, Esame*)
Esami(Codice, Materia, Data, Voto, Lode)
- Studenti(Nome, Cognome, Matricola, Anno, Esame*)
Esami(Codice, Materia, Data, Voto, Lode)

Esempio: altre soluzioni

14

- Studenti(Nome, Cognome, Matricola, Anno, Esame*)
Esami(Codice, Materia, Data, Voto, Lode)
- Studenti(Nome, Cognome, Matricola, Anno, Esame*)
Esami(Codice, Materia, Data, Voto, Lode)
- Studenti(Nome, Cognome, Matricola, Anno)
Esami(Codice, Materia, Data, Voto, Lode)
StudentiEsami(Esame*, Candidato*)

Esempio: altre soluzioni

14

- Studenti(Nome, Cognome, Matricola, Anno, Esame*)
Esami(Codice, Materia, Data, Voto, Lode)
- Studenti(Nome, Cognome, Matricola, Anno, Esame*)
Esami(Codice, Materia, Data, Voto, Lode)
- Studenti(Nome, Cognome, Matricola, Anno)
Esami(Codice, Materia, Data, Voto, Lode)
StudentiEsami(Esame*, Candidato*)
- Studenti(Nome, Cognome, Matricola, Anno)
Corsi(Materia, Crediti)
Esami(Codice, Corso*, Candidato*, Data, Voto, Lode)

- Studenti(Nome, Cognome, Matricola, Anno, Esame*)
Esami(Codice, Materia, Data, Voto, Lode)
- Studenti(Nome, Cognome, Matricola, Anno, Esame*)
Esami(Codice, Materia, Data, Voto, Lode)
- Studenti(Nome, Cognome, Matricola, Anno)
Esami(Codice, Materia, Data, Voto, Lode)
StudentiEsami(Esame*, Candidato*)
- Studenti(Nome, Cognome, Matricola, Anno)
Corsi(Materia, Crediti)
Esami(Codice, Corso*, Candidato*, Data, Voto, Lode)

● Quali sono sensate?

- Un attributo può avere valore non specificato (proprietà parziali), per varie ragioni:
 - non applicabile
 - sconosciuto
- si usa **NULL**
- Es.: Per lo schema di relazione nella biblioteca
Utente(Nome, Cognome, CodiceFiscale, ...)
CodiceFiscale per un ospite potrebbe non aver valore perché nel paese di provenienza il CF non si usa o perché il CF non cioè noto nel momento della creazione dell'utente.

- Negli schemi relazionali si può imporre il vincolo **not-null** per un attributo
- Gli attributi della **chiave primaria** (e delle chiavi in generale) devono assumere valori **non nulli**
- Una **chiave esterna** può avere valore nullo se rappresenta una associazione parziale.



<<UNIQUE(Materia, Candidato)>>

<<KEY(Materia, Candidato)>>

Dal Modello a Oggetti al Modello Relazionale

- Trasformazione per passi:
 - associazioni molti a uno (e uno a uno)
 - associazioni molti a molti
 - gerarchie di inclusione
 - identificazione chiavi primarie
 - attributi composti

Schemi a oggetti -> Schemi relazionali

20

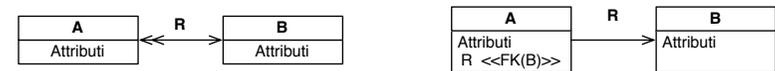
- Associazioni N:1 (univoche)



Schemi a oggetti -> Schemi relazionali

20

- Associazioni N:1 (univoche)



- totalità di R con vincolo not-null sulla chiave esterna

Associazioni N:1 (univoche)



- **totalità di R con vincolo not-null sulla chiave esterna**
- eventuali **attributi dell'associazione** si possono inserire in **A**

Associazioni N:1 (univoche)



- **totalità di R con vincolo not-null sulla chiave esterna**
- eventuali **attributi dell'associazione** si possono inserire in **A**
- **totalità dell'inversa non rappresentabile**

Associazioni N:1 (univoche)



- **totalità di R con vincolo not-null sulla chiave esterna**
- eventuali **attributi dell'associazione** si possono inserire in **A**
- **totalità dell'inversa non rappresentabile**

• Prestiti <<-|---->> Utenti

• EsamiEsterni <<-|---|-> EsamiInterni (attributo: Colloquio)

Associazioni 1:1 (univoche con inversa univoca)



● Associazioni 1:1 (univoche con inversa univoca)



- univocità di R^{-1} (inversa di R) con vincolo di chiave (<<unique>>, <<key>>) sulla chiave esterna
- possibile solo se R è totale
- la direzione di R scelta in modo che sia totale, se possibile

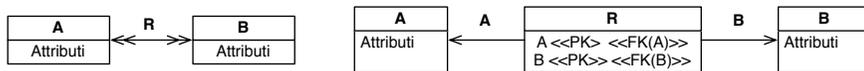
● Associazioni 1:1 (univoche con inversa univoca)



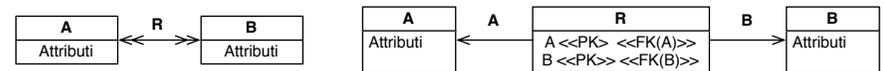
- univocità di R^{-1} (inversa di R) con vincolo di chiave (<<unique>>, <<key>>) sulla chiave esterna
- possibile solo se R è totale
- la direzione di R scelta in modo che sia totale, se possibile

● Es.: Domande Trasferimento <---|> Pratiche Trasferimento

● Associazioni N:M (multivalore con inversa multivalore)

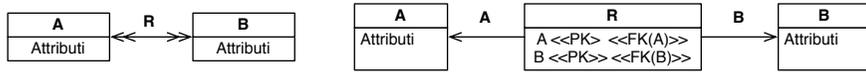


● Associazioni N:M (multivalore con inversa multivalore)



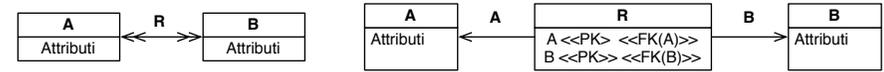
- eventuali attributi dell'associazione si inseriscono in R (e possono far parte della chiave primaria)

● Associazioni N:M (multivalore con inversa multivalore)



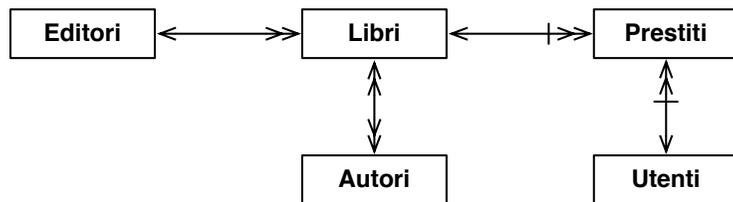
- eventuali attributi dell'associazione si inseriscono in R (e possono far parte della chiave primaria)
- totalità non rappresentabile

● Associazioni N:M (multivalore con inversa multivalore)

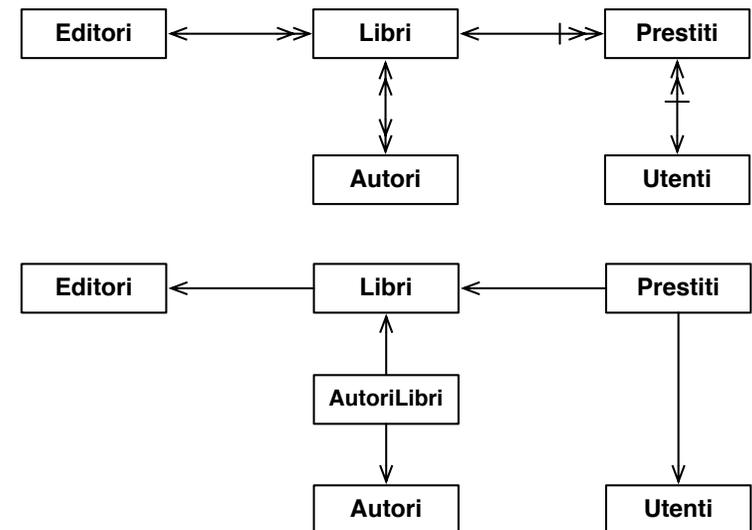


- eventuali attributi dell'associazione si inseriscono in R (e possono far parte della chiave primaria)
- totalità non rappresentabile
- Es.:
Relazione HaSostenuto, con attributo Data
TipoEsami <----> Studenti

Trasformazione di Schemi a Oggetti in Relazionali 23



Trasformazione di Schemi a Oggetti in Relazionali 23



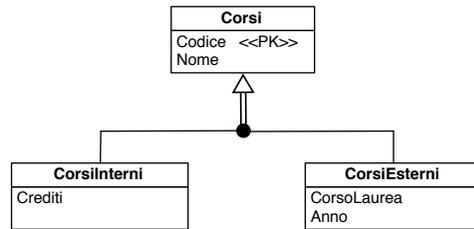
- Data la classe A (attr. X_A , chiave K_A) con sottoclassi B (attr. X_B) e C (attr. X_C)
- Tre possibili soluzioni

- Data la classe A (attr. X_A , chiave K_A) con sottoclassi B (attr. X_B) e C (attr. X_C)
- Tre possibili soluzioni
 - Relazione unica
 - $R(X_A, X_B, X_C, Discr)$
 - Discr indica la classe alla quale appartiene l'elemento
 - X_B e X_C possono avere valore nullo

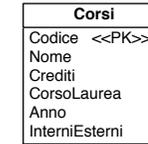
- Data la classe A (attr. X_A , chiave K_A) con sottoclassi B (attr. X_B) e C (attr. X_C)
- Tre possibili soluzioni
 - Relazione unica
 - $R(X_A, X_B, X_C, Discr)$
 - Discr indica la classe alla quale appartiene l'elemento
 - X_B e X_C possono avere valore nullo
 - Partizionamento verticale
 - $R_A(X_A)$: tutti gli elementi di A,
 - $R_B(X_B, K_A)$: attributi propri per gli elementi di B
 - $R_C(X_C, K_A)$: attributi propri per gli elementi di C

- Partizionamento orizzontale
 - $R_A(X_A)$: solo gli elementi di A - ($B \cup C$)
 - $R_B(X_A, X_B)$: elementi di B (tutti gli attributi)
 - $R_C(X_A, X_C)$: elementi di C (tutti gli attributi)

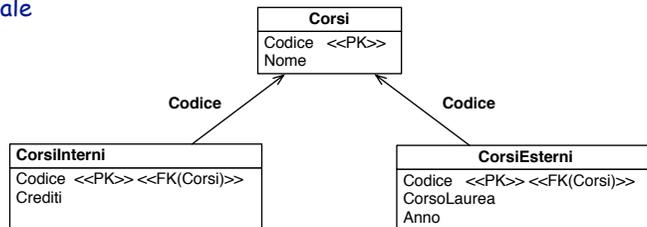
Si consideri la gerarchia seguente:



L'attributo **InterniEsterni** svolge il ruolo di discriminatore



Verticale



Orizzontale



Tabella unica

- conveniente se le sottoclassi differiscono per pochi attributi

Partizionamento orizzontale

- complica la visita di tutti gli elementi della superclasse
- divide la superclasse in più relazioni: sconsigliato se vi è una associazione entrante nella superclasse
- problematico senza vincolo di disgiunzione

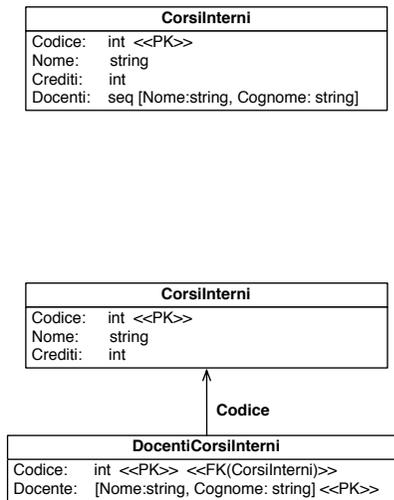
Partizionamento verticale

- complica il recupero di tutte le informazioni relative ad un'entità (distribuite in varie relazioni)

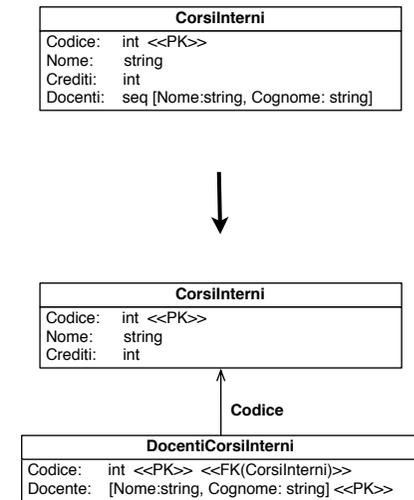
- Modello relazionale
 - relazione + { chiave primaria + chiave esterna + not null }
- Dal modello concettuale (a oggetti) al modello logico relazionale
 - associazioni A <<-> B (N:1 oppure 1:1) con chiave esterna in A
 - associazioni A <<-> B (M:N) con una nuova relazione R che riferisce con chiave esterna sia A che B
 - sottoclassi: relazione unica, partizionamento (verticale, orizzontale)
 - attributi multivalore e strutturati

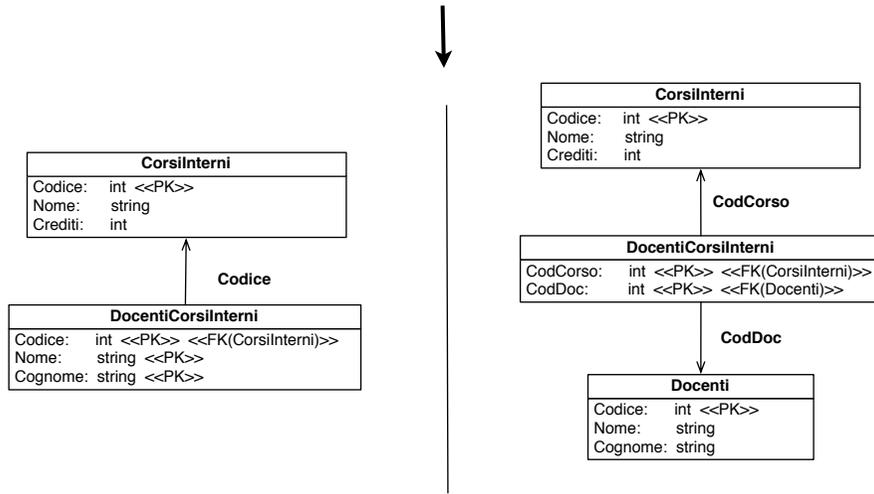
- Relazioni corrispondenti a **classi radice** (prive di superclasse)
 - attributo univoco, totale, costante
 - attributo artificiale (chiave sintetica)
- Relazioni che corrispondono a **sottoclassi**
 - chiave della superclasse
- Relazioni per **associazioni N:M**
 - concatenazione delle chiavi esterne

Attributi Multivalore e Strutturati



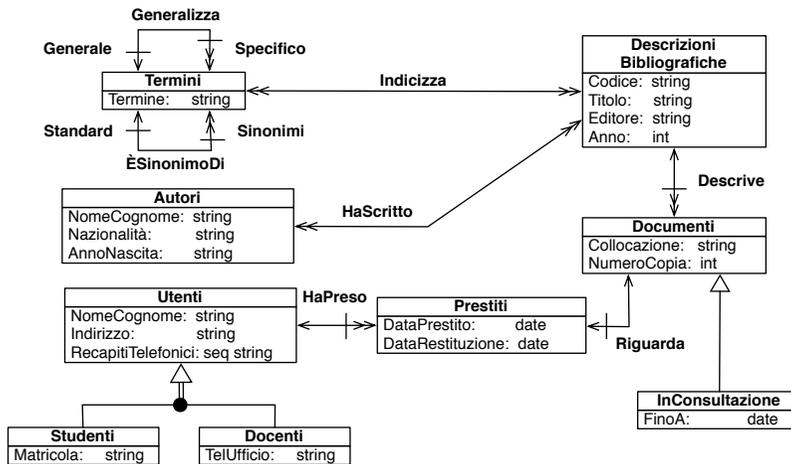
Attributi Multivalore e Strutturati



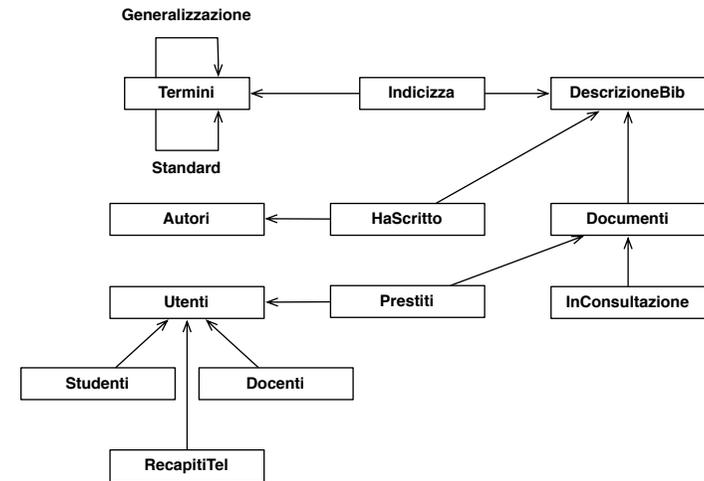


Un esempio: BD per una Biblioteca

Modello concettuale



Schema Logico (solo relazioni)



Schema delle relazioni (attributi, tipi, vincoli)

37

- Termini (Termine: string, Generalizzazione*: string, Standard*: string)
 - PK(Termine)
 - Generalizzazione FK(Termini), Standard FK(Termini)
- DescrizioneBib (Codice: string, Titolo: string, Editore: string, Anno: year)
 - PK(Codice)
- Indicizza (Termine*: string, Codice*: string)
 - PK(Termine, Codice)
 - Termine FK(Termini), Codice FK(DescrizioniBib)
- Autori (Codice: string, NomeCognome: string, Nazionalita: string, AnnoNascita: year)
 - PK(Codice)
- HaScritto (CodAutore*: string, CodDescr*: string)
 - PK(CodAutore, CodDescr)
 - CodAutore FK(Autori), CodDescr FK(DescrizioniBib)

Schema delle relazioni (Cont.)

38

- Documenti (Collocazione: string, NumeroCopia: int, CodDesc*: string)
 - PK(Collocazione)
 - CodDesc FK(DescrizioniBib) NOT NULL
- InConsultazione (Collocazione*: string, FinoA: string)
 - PK(Collocazione)
 - Collocazione FK(Documenti)
- Utenti (Codice: string, NomeCognome: string, Indirizzo: string)
 - PK(Codice)
- RecapitiTel (Numero: string, Utente*: string)
 - PK(Numero, Utente)
 - Utente FK(Utenti)

Tuesday, April 29, 2014

Schema delle relazioni (Cont.)

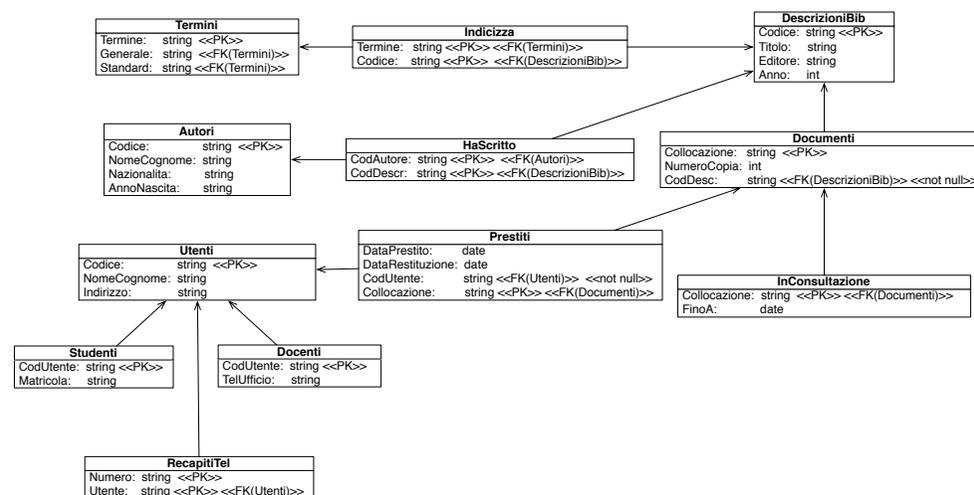
39

- Studenti (CodUtente*: string, Matricola: string)
 - PK(CodUtente)
 - CodUtente FK(Utenti)
- Docenti (CodUtente*: string, TelUfficio: string)
 - PK(CodUtente)
 - CodUtente FK(Utenti)
- Prestiti (DataPrestito: data, DataRestituzione: data, CodUtente*: string, Collocazione: string)
 - PK(Collocazione)
 - CodUtente FK(Utenti) NOT NULL, Collocazione FK(Documenti)

Tuesday, April 29, 2014

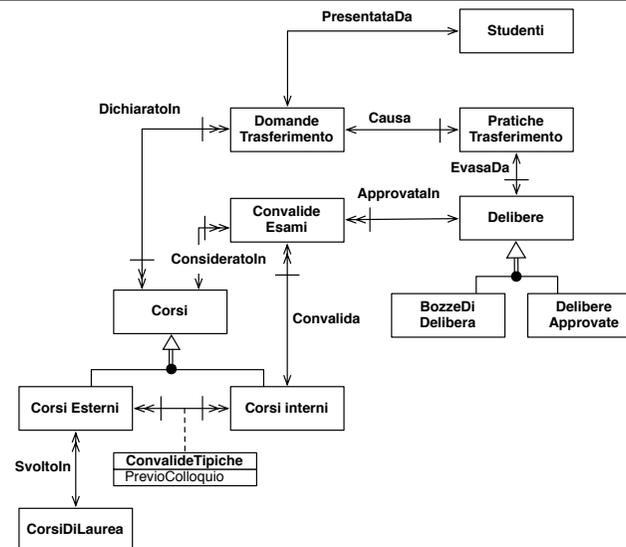
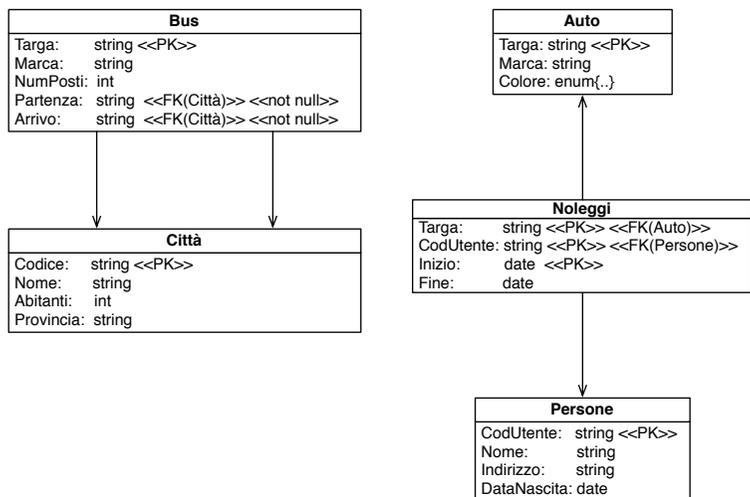
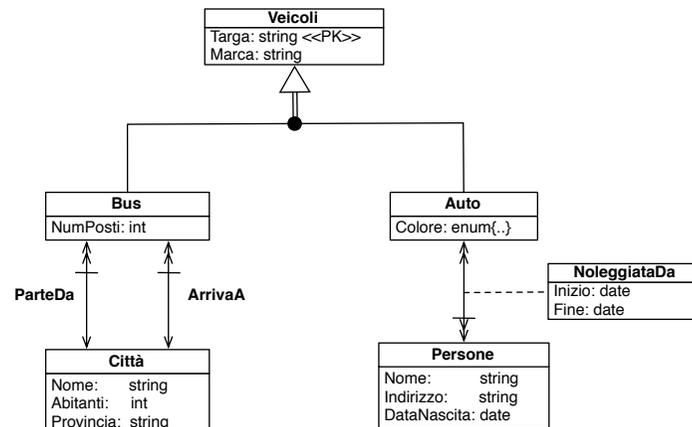
Schema relazionale con attributi, tipi, vincoli

40

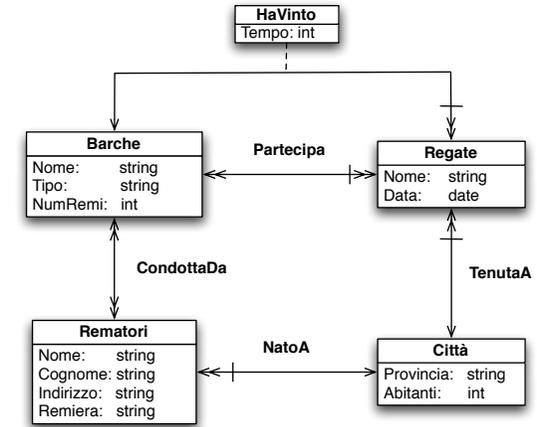


Tuesday, April 29, 2014

Si vogliono gestire informazioni relative ad una compagnia di trasporti. I veicoli a disposizione sono autobus e automobili. Un autobus effettua un viaggio collegando due città. Di ogni autobus interessa la marca, la targa, il numero dei posti, la città di partenza e la città di arrivo del suo tragitto. Delle automobili interessa la targa, la marca e il colore. Le automobili possono essere noleggiate da una persona di cui interessa l'età, il nome e l'indirizzo. Una persona può noleggiare più automobili e un'automobile può essere presa in affitto più volte. Di un affitto interessa il periodo. Infine di una città interessa il nome, il numero di abitanti e la provincia in cui si trova. Si dia uno schema grafico a oggetti della base di dati e si trasformi nello schema relazionale mostrandone la rappresentazione grafica. Poi per ogni relazione si specifichino attributi, chiave primaria e chiavi esterne.

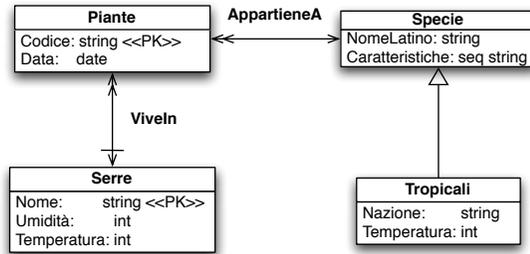


- Si vogliono gestire informazioni relative alle regate. Di una regata interessa il nome, la data e la città in cui si svolge e le barche partecipanti. Di una barca si vuole conoscere il nome della barca, il tipo di imbarcazione, il numero dei remi, l'equipaggio. Dei rematori interessa il nome e cognome, l'indirizzo, la remiera di appartenenza, il luogo di nascita. Per le città interessa la provincia e il numero di abitanti. Per ogni regata si vuol inoltre conoscere la barca vincitrice e il tempo impiegato per la regata.
- Si dia uno schema grafico a oggetti della base di dati e si trasformi nello schema relazionale mostrandone la rappresentazione grafica. Poi per ogni relazione si specifichi il nome degli attributi, la chiave primaria e le chiavi esterne.



- Si vogliono gestire informazioni relative alle piante di un orto botanico. Ogni pianta è identificata univocamente da un codice e si vuole conoscere la specie a cui appartiene e la data in cui è stata acquisita dall'orto botanico. Di una specie interessa conoscere il nome latino e la descrizione delle caratteristiche.
- Le piante possono essere tenute all'interno di serre. Di una serra interessa il nome, le specie di piante che contiene, l'umidità e la temperatura.
- Fra le piante un trattamento particolare è riservato alle piante tropicali. Di tali piante si vuole conoscere la nazione da dove provengono e la temperatura migliore a cui devono vivere.

- Si dia uno schema grafico a oggetti della base di dati e si trasformi nello schema relazionale mostrandone la rappresentazione grafica. Poi per ogni relazione si specifichi il nome degli attributi, la chiave primaria e le chiavi esterne.



Linguaggi Relazionali

Linguaggi Relazionali

- **Algebra Relazionale:** insieme di operatori su relazioni che danno come risultato relazioni; si definiscono
 - operatori primitivi (ridenominazione, proiezione, unione e differenza, restrizione, prodotto)
 - operatori derivati (giunzioni, divisione, ...)
 - altri operatori (raggruppamento, order by, min, max)
- Non si usa come direttamente come linguaggio di interrogazione dei DBMS ma come rappresentazione interna delle interrogazioni.
- **Calcolo Relazionale:** linguaggio dichiarativo di tipo logico dal quale è stato derivato l'SQL.

Algebra relazionale

• Data una **relazione** $R(A_1: T_1, \dots, A_n: T_n)$

• **Tipo:** $\{ A_1: T_1, \dots, A_n: T_n \}$

• **Grado:** n

• Data una ennupla $t \in R$

$t.A_i$ valore dell'attributo A_i

• Nel modello di base:

• relazioni come **insiemi** di ennuple

• **non** si usa **NULL**

• **Ridenominazione (δ)**

Data una relazione $R(X)$, con X insieme di attributi, $A \in X$ e $B \notin X$

$$\delta_{A \rightarrow B}(R)$$

relazione R dove A è ridenominato con B

$$\delta_{A \rightarrow B}(R) = \{t \mid \exists u \in R. t.B = u.A \wedge \forall C \in X - \{A\}. t.C = u.C\}$$

• Grado della nuova relazione? Tipo? Cardinalità?

R e S relazioni dello stesso tipo:

• **Unione (\cup)**

$$R \cup S$$

• **Differenza ($-$)**

$$R - S$$

• Qual è il tipo del risultato? Quante ennuple contiene il risultato?

• Se t_1 è un'ennupla non in R , allora

$$R = (R \cup \{t_1\}) - \{t_1\}$$

• **Proiezione (π):** data $R(X)$ con $\{A_1, \dots, A_n\} \subseteq X$

$$\pi_{A_1, A_2, \dots, A_n}(R)$$

"elimina" gli attributi diversi da A_1, \dots, A_n

$$\pi_{A_1, \dots, A_n}(R) = \{\langle t.A_1, \dots, t.A_n \rangle \mid t \in R\}$$

• **Proprietà:** se L_1 e L_2 sono liste di attributi

$$\pi_{L_1}(\pi_{L_2}(R)) = \pi_{L_1}(R)$$

• Qual è il tipo del risultato? Se R contiene n ennuple quante ne contiene il risultato?

- Sia data la relazione Studenti

Studenti	Nome	Cognome	Matricola	Anno	Prov
	Paolo	Verdi	71523	2005	PD
	Anna	Rossi	76366	2006	VE
	Giorgio	Zeri	71347	2005	PD
	Chiara	Scuri	71346	2006	PD

- Trovare il nome, la matricola e la provincia degli studenti

- Espressione nell'algebra

π Nome, Matricola, Provincia(Studenti)

Nome	Matricola	Provincia
Paolo	71523	PD
Anna	76366	VE
Giorgio	71347	PD
Chiara	71346	PD

π Provincia(Studenti) ?

Algebra Relazionale: Restrizione

- **Restrizione (selezione) (σ)**

$\sigma_{\text{Condizione}}(R)$

relazione le cui ennuple sono le ennuple di R che soddisfano la **Condizione**

- Condizione è una combinazione proposizionale di (dis)uguaglianze e disequazioni tra attributi (o tra attributi e costanti)

$\phi ::= A_i \text{ op } A_j \mid A_i \text{ op } c \mid \neg \phi \mid \phi \wedge \phi \mid \phi \vee \phi$

dove **op** è = oppure \leq

- La condizione riguarda attributi di **singole ennuple**

Algebra Relazionale: Restrizione (cont.)

- **Commutativa:**

$$\sigma_{C_1}(\sigma_{C_2}(R)) = \sigma_{C_1 \wedge C_2}(R) = \sigma_{C_2}(\sigma_{C_1}(R))$$

- Qual è il tipo del risultato? Se R contiene n ennuple quante ne ha il risultato?

- Trovare i dati degli studenti della provincia di Padova:

σ Provincia = 'PD' (Studenti)

Nome	Cognome	Matricola	Anno	Prov
Paolo	Verdi	71523	2005	PD
Giorgio	Zeri	71347	2005	PD
Chiara	Scuri	71346	2006	PD

- Trovare il nome, la matricola e l'anno di iscrizione degli studenti di Padova:

π Nome, Matricola, Anno (σ Provincia = 'PD' (Studenti))

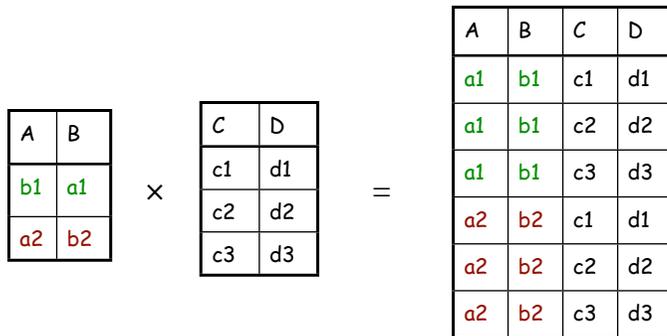
Nome	Matricola	Anno
Paolo	71523	2005
Giorgio	71347	2005
Chiara	71346	2006

- Prodotto (\times)

$$R \times S$$

- R e S con attributi **distinti** A_1, \dots, A_n , e B_1, \dots, B_m
- ennuple ottenute concatenando ennuple di R e ennuple di S
- $R \times S = \{ \langle t.A_1, \dots, t.A_n, u.B_1, \dots, u.B_m \rangle \mid t \in R \wedge u \in S \}$

- Qual è il tipo del risultato? Se R e S contengono n e m ennuple quante ne contiene il risultato? Chiave?



- Qual è il risultato di **Studenti** \times **Esami** ?

Studenti

Nome	Cognome	Matricola	Anno	Provincia
Paolo	Verdi	71523	2005	PD
Anna	Rossi	76366	2006	VE
Giorgio	Zeri	71347	2005	PD
Chiara	Scuri	71346	2006	PD

Esami

Codice	Materia	Candidato*	Data	Voto	Lode
B112	BD1	71523	08.07.06	27	N
F31	FIS	76366	08.07.07	26	N
B247	BD2	71523	28.12.06	30	S

- Trovare il nome degli studenti che hanno superato l'esame di BD con 30

$$\pi_{\text{Nome}}(\sigma_{\text{Materia}='BD' \wedge \text{Voto}=30}(\sigma_{\text{Matricola}=\text{Candidato}}(\text{Studenti} \times \text{Esami})))$$

- si introduce un operatore derivato: la giunzione!

$$\pi_{\text{Nome}}(\sigma_{\text{Materia} = 'BD' \wedge \text{Voto} = 30}(\text{Studenti} \bowtie_{\text{Matricola}=\text{Candidato}} \text{Esami}))$$

- **Giunzione**: Utile per "combinare" informazioni di relazioni correlate

$$R \bowtie_{A_i=B_j} S$$

- R e S con attributi distinti $A_1, \dots, A_n, e B_1, \dots, B_m$

- ovvero

$$R \bowtie_{A_i=B_j} S = \sigma_{A_i=B_j}(R \times S)$$

$$\{(t.A_1, \dots, t.A_n, u.B_1, \dots, u.B_m) \mid t \in R \wedge u \in S \wedge t.A_i = u.B_j\}$$

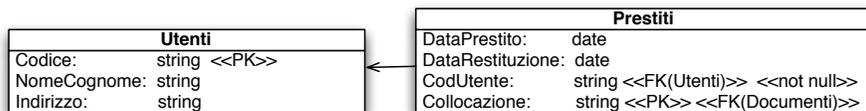
- **Giunzione naturale**

$$R \bowtie S$$

Esempio

- Giunzione

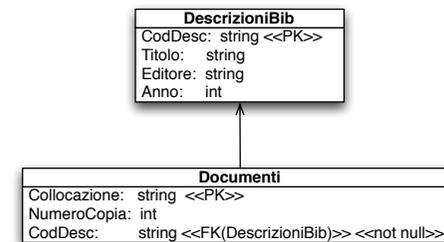
Utenti $\bowtie_{\text{Codice}=\text{CodUtente}}$ Prestiti



Esempio

- Giunzione naturale

Documenti \bowtie DescrizioniBib



● **Intersezione**

$$R \cap S$$

esprimibile come

$$R - (R - S)$$

- **Divisione:** date le relazioni $R(XY)$ e $S(Y)$ si vuole produrre una relazione $T(X)$ tale che una ennupla t è in T se e solo se **per ogni** s in S la ennupla $\langle t, s \rangle$ appare in R .

$$R \div S$$

- **Esempio:** matricola degli studenti che hanno fatto tutti gli esami che ha fatto Anna Rossi (matr. 76366).

- esami di Anna Rossi:

$$ES_AR = \pi_{Materia}(\sigma_{Matricola='76366'}(\text{Studenti} \bowtie_{Candidato=Matricola} \text{Esami}))$$

- esami studenti con matricola

$$ES = \pi_{Matricola, Materia}(\text{Studenti} \bowtie_{Candidato=Matricola} \text{Esami})$$

Operatori derivati: Divisione (cont.)

- il risultato desiderato è quindi

$$ES \div ES_AR$$

- Usato per query che coinvolgono quantificazione universale

- Esprimibile come

$$\pi_X(R) - \pi_X((\pi_X(R) \times S) - R)$$

Esercizio

- Query per
 - studenti che hanno fatto un sottoinsieme degli esami di Anna Rossi
 - studenti che hanno fatto esattamente gli esami di Anna Rossi

● **Proiezione generalizzata**

$$\pi_{Exp_1 AS A_1, Exp_2 AS A_2, \dots, Exp_n AS A_n}(R)$$

● Le espressioni Exp_i possono comprendere attributi, costanti, e operazioni su di essi

● **Esempio:** data una relazione $Utente(Codice, SalarioLordo, Trattenute, \dots)$

$$\pi_{Codice, SalarioLordo - Trattenute AS Stipendio}(Utente)$$

● **Proiezione senza duplicati (multinsiemistica)**

$$\pi_{A_1, A_2, \dots, A_n}^b(R)$$

● **Ordinamento (liste di ennuple)**

$$\tau_{A_1, A_2, \dots, A_n}(R)$$

Altri Operatori (cont.)

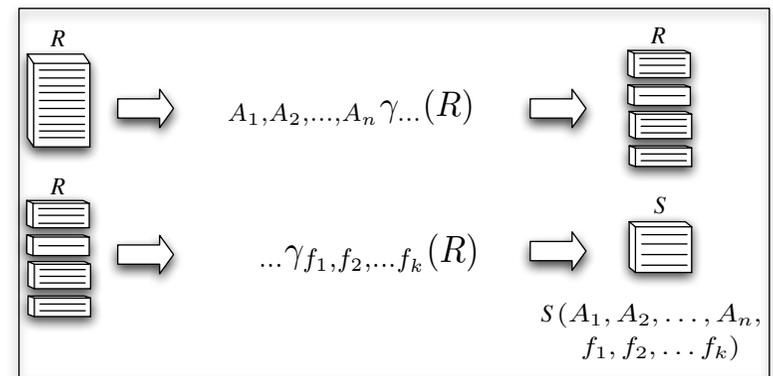
● **Raggruppamento (γ)**

$$A_1, A_2, \dots, A_n \gamma_{f_1, f_2, \dots, f_k}(R)$$

dove gli A_i sono attributi di R e le f_i sono espressioni che usano funzioni di aggregazione (min, max, count, sum, avg, ...)

Significato del raggruppamento (γ)

$$S = A_1, A_2, \dots, A_n \gamma_{f_1, f_2, \dots, f_k}(R)$$



- Trovare per ogni candidato il numero degli esami, il voto minimo, massimo e medio

$\{\text{Candidato}\} \gamma_{\{\text{count}(*), \text{min}(\text{Voto}), \text{max}(\text{Voto}), \text{avg}(\text{Voto})\}}(\text{Esami})$

Materia	Candidato	Data	Voto	Lode
BD1	71523	08.07.06	20	N
FIS	76366	08.07.07	26	N
DA	71523	28.12.06	30	S
BD2	76366	28.12.06	28	S

- raggruppamento

Materia	Candidato	Data	Voto	Lode
BD1	71523	08.07.06	20	N
DA	71523	28.12.06	30	S
FIS	76366	08.07.07	26	N
BD2	76366	28.12.06	28	S

- raggruppamento

Materia	Candidato	Data	Voto	Lode
BD1	71523	08.07.06	20	N
DA	71523	28.12.06	30	S
FIS	76366	08.07.07	26	N
BD2	76366	28.12.06	28	S

- calcolo delle funzioni

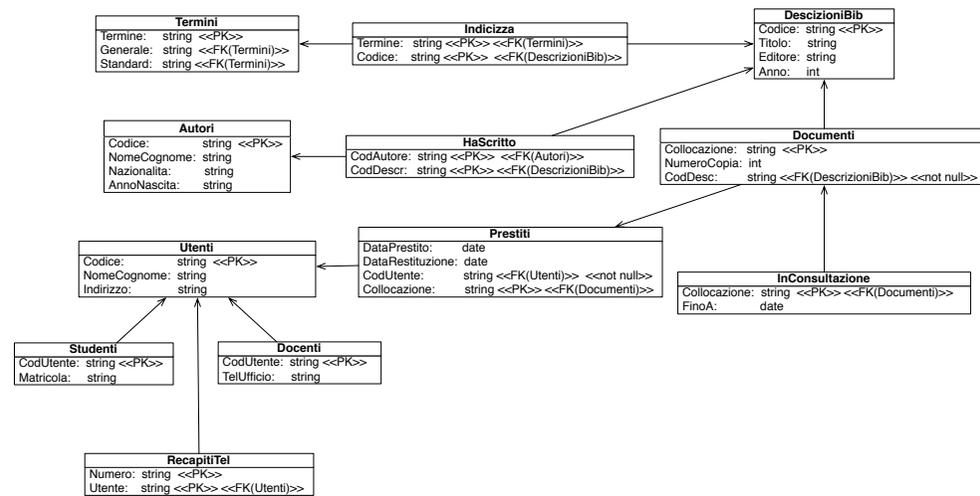
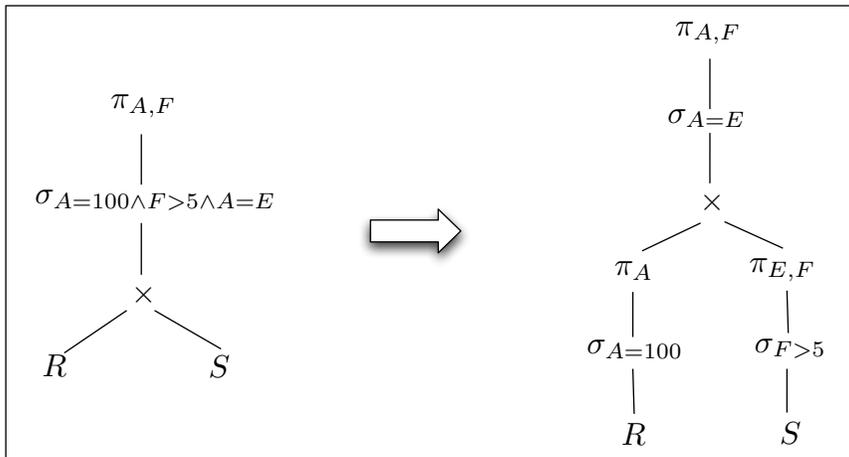
Candidato	Count(*)	min(Voto)	max(Voto)	avg(Voto)
71523	2	20	30	25
76366	2	26	28	27

- Basate su regole di equivalenza fra espressione algebriche
- Consentono di scegliere diversi ordini di join e di anticipare proiezioni e restrizioni.
- Alcuni esempi con la relazione $R(A, B, C, D)$:

$$\begin{aligned} \pi_A(\pi_{A,B}(R)) &\equiv \pi_A(R) \\ \sigma_{C_1}(\sigma_{C_2}(R)) &\equiv \sigma_{C_1 \wedge C_2}(R) \\ \sigma_{C_1 \wedge C_2}(R \times S) &\equiv \sigma_{C_1}(R) \times \sigma_{C_2}(S) \\ R \times (S \times T) &\equiv (R \times S) \times T \\ (R \times S) &\equiv (S \times R) \\ \sigma_C(X \gamma_F(R)) &\equiv_X \gamma_F(\sigma_C(R)) \end{aligned}$$

Consideriamo le relazioni $R(A, B, C, D)$ e $S(E, F, G)$ e l'espressione:

$$\pi_{A,F}(\sigma_{A=100 \wedge F > 5 \wedge A=E}(R \times S))$$



Query sulla biblioteca

- Titolo e collocazione di tutti i documenti in prestito
- Nome e Cognome degli utenti che hanno documenti in prestito
- Codice degli utenti che hanno in prestito tutti i libri (almeno un documento per ogni descrizione bibliografica)
- Codice degli utenti che hanno in prestito tutti i libri di fisica (si legga "libro di fisica" come documento la cui descrizione bibliografica è indicizzata da un termine che è "Fisica" oppure ha come standard "Fisica").
- Codice degli utenti che hanno in prestito solo libri di fisica
- Codice degli utenti che hanno in prestito tutti e soli i libri di fisica

Query sulla biblioteca (cont.)

- Tutti gli utenti che:
 - sono studenti e hanno matricola < 7000
 - sono docenti e hanno numero di telefono tra 1300 e 1500
- Gli utenti che non hanno in prestito nessun libro
- Gli utenti che hanno in prestito più di tre libri
- Gli autori che hanno scritto il massimo numero di libri

Calcolo Relazionale

Calcolo relazionale

85

- L'algebra relazionale non è l'unico linguaggio formale di interrogazione per DB relazionali; un'alternativa è il **calcolo relazionale (CR)**, del quale esistono due varianti:
 - **calcolo relazionale su ennuple (CRE)**
 - **calcolo relazionale su domini (CRD)**

Completezza Relazionale

86

- AR, CRE e CRD sono **espressivamente equivalenti**: ogni interrogazione esprimibile nell'uno è anche esprimibile negli altri.
- Un linguaggio relazionale espressivamente equivalente all'AR, al CRE e al CRD è detto **relazionalmente completo**
- i linguaggi dei DBMSs commerciali sono in genere non solo relazionalmente completi, ma anche di più ... in quanto includono anche altre funzionalità (e.g. aggregazione, raggruppamento, ...).

Differenze fra AR e CR

87

- AR è un linguaggio **procedurale**
 - un'interrogazione è una espressione che specifica, oltre a cosa va recuperato, le operazioni necessarie a recuperarlo;
- CR è un linguaggio **dichiarativo**
 - un'interrogazione è un'espressione che specifica cosa va recuperato, ma non come recuperarlo.
 - le operazioni da eseguire e la loro sequenzializzazione sono decise dal DBMS.
- Praticamente tutti i linguaggi dei DBMS relazionali commerciali sono implementazioni (più o meno fedeli ...) del CR; ad esempio SQL ~ CRE

- termini: denotano individui (elementi del dominio di interesse)

$$t ::= c \mid x \mid f(t_1, \dots, t_n)$$

- c costante
- x variabile
- f simbolo di funzione

- formule: denotano valori di verità (T o F):

$$\phi ::= p(t_1, \dots, t_n) \mid \neg\phi \mid \phi_1 \wedge \phi_2 \mid \phi_1 \rightarrow \phi_2 \mid \forall x.\phi \mid \exists x.\phi$$

- p simbolo di predicato n-ario

- Il CRE usa la logica del prim'ordine, interpretata su un dominio i cui elementi sono le ennuple della BD, per esprimere le interrogazioni

- costanti e le variabili sono di tipo ennupla.

- Esempio di interrogazione:

Nomi e cognomi degli studenti che hanno superato almeno un esame:

$$\{t.Nome, t.Cognome \mid t \in Studenti \wedge \exists e \in Esami.(t.Matricola = e.Candidato)\}$$

- Un'interrogazione del CRE è un'espressione del tipo

$$\{ t_{i1}.A_1, \dots, t_{im}.A_m \mid \Phi(t_1, \dots, t_n) \}$$

dove

- t_i variabili ennupla (il cui tipo, i.e. a quali relazioni appartengono, sarà indicato in Φ);
- A_i simboli di funzione di tipo attributo ($t_i.A_i$ è una notazione alternativa per $A_i(t_i)$);
- $\Phi(t_1, \dots, t_n)$ è una formula del prim'ordine in cui
 - le variabili t_1, \dots, t_n occorrono libere
 - il risultato è l'insieme delle ennuple $\langle t_{i1}.A_1, \dots, t_{im}.A_m \rangle$ tali che $\Phi(t_1, \dots, t_n)$ è vera.

- Le formule atomiche possono essere

- formule di tipo

$$t \in Studenti \quad e \in Esami$$

dichiara che t appartiene all'estensione corrente di Studente: quindi in ogni espressione t.A nell'interrogazione, A deve essere un'attributo di Studente;

- formule di confronto fra valori di attributi

$$t.Matricola = e.Candidato$$

- formule di confronto fra il valore di un attributo e un valore costante

$$t.Provincia = 'PD'$$

● Selezione

$\sigma_{\text{Prov}='PD'}(\text{Studente})$

$\{ t \mid t \in \text{Studente} \wedge t.\text{Provincia}='PD' \}$

● Proiezione

$\pi_{\text{Nome,Cognome}}(\text{Studente})$

$\{ t.\text{Nome}, t.\text{Cognome} \mid t \in \text{Studente} \}$

● Unione

$\text{Studenti} \cup \text{Docenti}$

$\{ t \mid t \in \text{Studenti} \vee t \in \text{Docenti} \}$

● Differenza

$\text{Studenti} - \text{Docenti}$

$\{ t \mid t \in \text{Studenti} \wedge \neg (t \in \text{Docenti}) \}$

● Prodotto cartesiano

$\text{Studenti} \times \text{Esami}$

$\{ s, e \mid s \in \text{Studenti} \wedge e \in \text{Esami} \}$

● Intersezione

$\text{Studenti} \cap \text{Docenti}$

$\{ t \mid t \in \text{Studenti} \wedge t \in \text{Docenti} \}$

Esercizi

● Esprimere nel calcolo relazionale

- giunzione;
- giunzione naturale;
- divisione

● Vediamo ad esempio la divisione. Date le relazioni $R(A,B)$ e $S(B)$

Esercizi

● Esprimere nel calcolo relazionale

- giunzione;
- giunzione naturale;
- divisione

● Vediamo ad esempio la divisione. Date le relazioni $R(A,B)$ e $S(B)$

$$R \div S = \{ t.A \mid t \in R \wedge \forall s \in S. \exists t' \in R. (t'.A = t.A \wedge t'.B = s.B) \}$$