

Basi di Dati

Algebra Relazionale

Concetti Fondamentali

Concetti Fondamentali

- ◆ **Introduzione**
- ◆ **La Base di Dati di Esempio**
- ◆ **Algebra Relazionale**
 - selezione, proiezione
 - prodotto cartesiano, join
 - unione, intersezione, differenza
 - ridenominazioni
- ◆ **Forma Standard**

Introduzione

- ◆ **SQL**
 - Integra linguaggi per scopi diversi: DDL, DCL, DML
- ◆ **Linguaggio di Definizione (DDL)**
 - Creare la base di dati e le tabelle (schema)
- ◆ **Linguaggio di Controllo (DCL)**
 - Creare utenti e autorizzazioni
- ◆ **Linguaggio di Manipolazione (DML)**
 - Inserire, cancellare, modificare le entuple
 - Interrogare la base di dati

Introduzione

- ◆ **DDL, DCL**
 - concettualmente semplici
- ◆ **DML**
 - aggiornamenti: concettualmente semplice
 - interrogazioni: complesso
- ◆ **Lavorare con oggetti inconsueti**
 - le tabelle
 - algebra su tabelle; analogo: algebra sui reali

Introduzione

◆ Algebra per Tabelle

- insieme di operatori che applicati a tabelle restituiscono tabelle
 - analogo: $y+3$, $x-2$
- espressioni
 - analogo: $z=(y+3)-2$
- sintassi astratta

- semantica operativa astratta

Base di Dati dei Corsi di Informatica

◆ Professori

- codice, nome, qualifica, facoltà
- numeri di telefono

◆ Studenti

- matricola, nome, tipo di corso (ciclo): laurea triennale / specialistica
- relatore della tesi

◆ Corsi

- codice, titolo, docente, ciclo

◆ Esami

- studente, voto, lode, corso

◆ Tutorato Studentesco

- studente tutore, studente tutorato

```
TABLE Professori (
  cod char(4) PRIMARY KEY,
  cognome varchar(20) NOT NULL,
  nome varchar(20) NOT NULL,
  qualifica char(15),
  facolta char(10) );
```

```
TABLE Tutorato (
  studente integer
  REFERENCES Studenti(matr),
  tutor integer
  REFERENCES Studenti(matr),
  UNIQUE (studente,tutor));
```

```
TABLE Studenti (
  matr integer PRIMARY KEY,
  cognome varchar(20) NOT NULL,
  nome varchar(20) NOT NULL,
  ciclo char(20),
  anno integer,
  relatore char(4)
  REFERENCES Professori(cod)
);
```

```
TABLE Esami (
  studente integer
  REFERENCES Studenti(matr),
  corso char(3)
  REFERENCES Corsi(cod),
  voto integer,
  lode bool,
  CHECK (voto>=18 and voto<=30),
  CHECK (not lode or voto=30),
  UNIQUE (studente, corso));
```

```
TABLE Corsi (
  cod char(3) PRIMARY KEY,
  titolo varchar(20) NOT NULL,
  ciclo char(20),
  docente char(4)
  REFERENCES Professori(cod)
);
```

```
TABLE Numeri (
  professore char(4)
  REFERENCES Professori(cod),
  numero char(9),
  UNIQUE (professore,numero));
```

Professori

<u>cod</u>	cognome	nome	qualifica	facolta
FT	Totti	Francesco	ordinario	Ingegneria
CV	Vieri	Christian	associato	Scienze
ADP	Del Piero	Alessandro	supplente	null

Studenti

<u>matr</u>	cognome	nome	ciclo	anno	relatore
111	Rossi	Mario	laurea tr.	1	null
222	Neri	Paolo	laurea tr.	2	null
333	Rossi	Maria	laurea tr.	1	null
444	Pinco	Palla	laurea tr.	3	FT
77777	Bruno	Pasquale	laurea sp.	1	FT
88888	Pinco	Pietro	laurea sp.	1	CV

Corsi

<u>cod</u>	titolo	ciclo	docente
PR1	Programmazione I	laurea tr.	FT
ASD	Algoritmi e Str. Dati	laurea tr.	CV
INFT	Informatica Teorica	laurea sp.	ADP

Tutorato

studente	tutor
111	77777
222	77777
333	88888
444	88888

Numeri

professore	numero
FT	0971205145
FT	347123456
VC	0971205227
ADP	0971205363
ADP	338123456

Esami

studente	corso	voto	lode
111	PR1	27	false
222	ASD	30	true
111	INFT	24	false
77777	PR1	21	false
77777	ASD	20	false
88888	ASD	28	false
88888	PR1	30	false
88888	INFT	30	true

Algebra Relazionale

◆ Collezione di operatori

- applicati a tabelle
- risultato = ancora tabelle

◆ Espressioni

- composizione di operatori applicati a tabelle

◆ Assegnazioni

- consentono di assegnare ad una nuova tabella il risultato di un'espressione

Algebra Relazionale

◆ **Attenzione**

- lavoreremo con due tipi di tabelle: (1) della BD e (2) “temporanee”

◆ **(1) Tabelle originali della base di dati**

- con schema completo di vincoli

◆ **(2) Tabelle “temporanee”**

- risultato di interrogazioni dell'algebra
- per queste tabelle non viene definito uno schema vero e proprio
- ereditano parte dello schema (attributi e tipi) dall'interrogazione di cui sono il risultato

Algebra Relazionale

◆ **Operatori principali**

- selezione
- proiezione
- prodotto cartesiano e join
- unione, intersezione e differenza
- ridenominazione

Selezione

◆ Funzione

- serve per selezionare alcune delle ennuple di una tabella scartando le altre
- sulla base di una condizione

◆ Esempio

- “Estrarre dalla base di dati una tabella, *StudentiTriennio*, contenente i dati degli studenti della laurea triennale”

“Studenti della laurea triennale”

Studenti	<u>matr</u>	cognome	nome	ciclo	anno	relatore
	111	Rossi	Mario	laurea tr.	1	null
	222	Neri	Paolo	laurea tr.	2	null
	333	Rossi	Maria	laurea tr.	1	null
	444	Pinco	Palla	laurea tr.	3	FT
	77777	Bruno	Pasquale	laurea sp.	1	FT
	88888	Pinco	Pietro	laurea sp.	1	CV

condizione: Studenti.ciclo='laurea tr.'

StudentiTriennio	matr	cognome	nome	ciclo	anno	relatore
	111	Rossi	Mario	laurea tr.	1	null
	222	Neri	Paolo	laurea tr.	2	null
	333	Rossi	Maria	laurea tr.	1	null
	444	Pinco	Palla	laurea tr.	3	FT

Selezione: Sintassi

◆ Operatore unario (“monadico”)

- condizione sui valori degli attributi

◆ Condizione: espressione booleana

- operandi: valori degli attributi della tabella
- operatori di confronto, operatori booleani

◆ Sintassi: $\sigma_{\text{condizione}}(R)$

◆ Esempio di uso:

StudentiTriennio = $\sigma_{\text{ciclo}='laurea\ tr.'}(\text{Studenti})$

“Studenti della laurea tr. di anni successivi al 1°”

Studenti	<u>matr</u>	cognome	nome	ciclo	anno	relatore
	111	Rossi	Mario	laurea tr.	1	null
	222	Neri	Paolo	laurea tr.	2	null
	333	Rossi	Maria	laurea tr.	1	null
	444	Pinco	Palla	laurea tr.	3	FT
	77777	Bruno	Pasquale	laurea sp.	1	FT
	88888	Pinco	Pietro	laurea sp.	1	CV

condizione: ciclo='laurea tr.' AND anno > 1

Risultato	matr	cognome	nome	ciclo	anno	relatore
	222	Neri	Paolo	laurea tr.	2	null
	444	Pinco	Palla	laurea tr.	3	FT

Risultato = $\sigma_{\text{ciclo}='laurea\ tr.' \text{ AND } \text{anno} > 1}(\text{Studenti})$

Selezione: Semantica

◆ Schema del risultato

- attributi e tipi della tabella originale

◆ Istanza del risultato

- ennuple della tabella i cui valori soddisfano la condizione
- cardinalità minore o uguale rispetto alla tabella originale

Selezione: Semantica

◆ Nota sullo schema

- si tratta di una tabella temporanea
- i nomi (e i tipi) degli attributi sono ereditati dalle tabelle della base di dati

```
TABLE StudentiTriennio (  
  matr integer,  
  cognome varchar(20),  
  nome varchar(20),  
  ciclo char(20),  
  anno integer,  
  relatore char(4));
```

oppure

```
TABLE StudentiTriennio (  
  Studenti.matr integer,  
  Studenti.cognome varchar(20),  
  Studenti.nome varchar(20),  
  Studenti.ciclo char(20),  
  Studenti.anno integer,  
  Studenti.relatore char(4));
```

N.B.: “Studenti” è la tabella su cui è stata effettuata la selezione

Proiezione

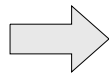
◆ Funzione

- estrarre alcune delle colonne di una tabella

◆ Esempio

- “Estrarre l’elenco dei nomi e i cognomi degli studenti”

```
TABLE Studenti (  
  matr integer,  
  cognome varchar(20),  
  nome varchar(20),  
  ciclo char(20),  
  anno integer,  
  relatore char(4));
```



```
TABLE ElencoNomi (  
  Studenti.cognome varchar(20),  
  Studenti.nome varchar(20),  
  );
```

“Cognomi e Nomi degli Studenti”

Studenti	<u>matr</u>	cognome	nome	ciclo	anno	relatore
	111	Rossi	Mario	laurea tr.	1	null
	222	Neri	Paolo	laurea tr.	2	null
	333	Rossi	Maria	laurea tr.	1	null
	444	Pinco	Palla	laurea tr.	3	FT
	77777	Bruno	Pasquale	laurea sp.	1	FT
	88888	Pinco	Pietro	laurea sp.	1	CV

ElencoNomi	cognome	nome
	Rossi	Mario
	Neri	Paolo
	Rossi	Maria
	Pinco	Palla
	Bruno	Pasquale
	Pinco	Pietro

Proiezione: Sintassi

◆ Operatore unario

- lista di attributi della tabella

◆ Sintassi

$$\pi_{\text{lista attributi}}(R)$$

◆ Esempio d'uso

$$\text{ElencoNomi} = \pi_{\text{cognome, nome}}(\text{Studenti})$$

Proiezione: Semantica

◆ Schema del risultato

- attributi dello schema originale su cui si effettua la proiezione

◆ Istanza del risultato

- restrizione (“proiezione”) delle ennuple originali agli attributi specificati

◆ ATTENZIONE

- se nel risultato non sopravvivono chiavi dello schema originale possono esserci duplicati

“Cognomi e Anni di Corso degli Studenti”

Studenti	matr	cognome	nome	ciclo	anno	relatore
	111	Rossi	Mario	laurea tr.	1	null
	222	Neri	Paolo	laurea tr.	2	null
	333	Rossi	Maria	laurea tr.	1	null
	444	Pinco	Palla	laurea tr.	3	FT
	77777	Bruno	Pasquale	laurea sp.	1	FT
	88888	Pinco	Pietro	laurea sp.	1	CV

Risultato

cognome	anno
Rossi	1
Neri	2
Rossi	1
Pinco	3
Bruno	1
Pinco	1

Risultato = $\pi_{\text{cognome, anno}}(\text{Studenti})$

ennuple identiche (duplicati)

NOTA: ulteriori duplicati se
Risultato' = $\pi_{\text{cognome}}(\text{Studenti})$

Prodotto Cartesiano

◆ Finora

- operatori unari
- lavorano sui dati di un'unica tabella

◆ In realtà

- è nella natura del modello relazionale frammentare i dati tra le tabelle
- molto spesso è necessario correlare dati provenienti da tabelle diverse
- è possibile utilizzare il prodotto cartesiano

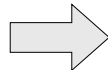
Prodotto Cartesiano

◆ Esempio

- “Generare la tabella *ProfessoriENumeri* contenente nomi, cognomi e numeri dei prof.”

```
TABLE Professori (
  cod char(4) PRIMARY KEY,
  cognome varchar(20) NOT NULL,
  nome varchar(20) NOT NULL,
  qualifica char(15),
  facolta char(10) );
```

```
TABLE Numeri (
  professore char(4)
  REFERENCES Professori(cod),
  numero char(9),
  PRIMARY KEY (professore, numero));
```



```
TABLE ProfessoriENumeri (
  Professori.cognome varchar(20),
  Professori.nome varchar(20),
  numero char(9));
```

“Professori e Numeri”

Professori

cod	cognome	nome	qualifica	facolta
FT	Totti	Francesco	ordinario	Ingegneria
CV	Vieri	Christian	associato	Scienze
ADP	Del Piero	Alessandro	supplente	null

Numeri

professore	numero
FT	0971205145
FT	347123456
CV	0971205227
ADP	0971205363
ADP	338123456

1° Passo: Prodotto Cartesiano

TabellaA = Professori X Numeri

cod	cognome	nome	qualifica	facolta	professore	numero
FT	Totti	Francesco	ordinario	Ingegneria	FT	0971205145
FT	Totti	Francesco	ordinario	Ingegneria	FT	347123456
FT	Totti	Francesco	ordinario	Ingegneria	CV	0971205227
FT	Totti	Francesco	ordinario	Ingegneria	ADP	0971205363
FT	Totti	Francesco	ordinario	Ingegneria	ADP	338123456
CV	Vieri	Christian	associato	Scienze	FT	0971205145
CV	Vieri	Christian	associato	Scienze	FT	347123456
...
ADP	Del Piero	Alessandro	supplente	null	ADP	338123456

“Professori e Numeri”

TabellaA

cod	cognome	nome	qualifica	facolta	professore	numero
FT	Totti	Francesco	ordinario	Ingegneria	FT	0971205145
FT	Totti	Francesco	ordinario	Ingegneria	FT	347123456
FT	Totti	Francesco	ordinario	Ingegneria	CV	0971205227
FT	Totti	Francesco	ordinario	Ingegneria	ADP	0971205363
FT	Totti	Francesco	ordinario	Ingegneria	ADP	338123456
CV	Vieri	Christian	associato	Scienze	FT	0971205145
...
ADP	Del Piero	Alessandro	supplente	null	ADP	338123456

2° Passo: Selezione $\text{TabellaB} = \sigma_{\text{cod}=\text{professore}}(\text{TabellaA})$

cod	cognome	nome	qualifica	facolta	professore	numero
FT	Totti	Francesco	ordinario	Ingegneria	FT	0971205145
FT	Totti	Francesco	ordinario	Ingegneria	FT	347123456
CV	Vieri	Christian	associato	Scienze	CV	0971205227
ADP	Del Piero	Alessandro	supplente	null	ADP	0971205363
ADP	Del Piero	Alessandro	supplente	null	ADP	338123456

“Professori e Numeri”

Tabella B

cod	cognome	nome	qualifica	facolta	professore	numero
FT	Totti	Francesco	ordinario	Ingegneria	FT	0971205145
FT	Totti	Francesco	ordinario	Ingegneria	FT	347123456
CV	Vieri	Christian	associato	Scienze	VC	0971205227
ADP	Del Piero	Alessandro	supplente	null	ADP	0971205363
ADP	Del Piero	Alessandro	supplente	null	ADP	338123456

3° Passo: Proiezione $\text{ProfessoriENumeri} = \pi_{\text{cognome, nome, numero}}(\text{TabellaB})$

cognome	nome	numero
Totti	Francesco	0971205145
Totti	Francesco	347123456
Vieri	Christian	0971205227
Del Piero	Alessandro	0971205363
Del Piero	Alessandro	338123456

Professori e Numeri

◆ In sintesi

TabellaA = Professori X Numeri

TabellaB = $\sigma_{\text{cod=professore}}$ (TabellaA)

ProfessoriENumeri = $\pi_{\text{cognome, nome, numero}}$ (TabellaB)

◆ Ovvero

ProfessoriENumeri = $\pi_{\text{cognome, nome, numero}} (\sigma_{\text{cod=professore}} (\text{Professori X Numeri}))$

Prodotto Cartesiano: Sintassi

◆ Operatore binario (“diadico”)

- è il primo operatore che consente di mettere assieme dati provenienti da tabelle diverse

◆ Sintassi

- R X S

Prodotto Cartesiano: Semantica

◆ Schema del risultato

- unione degli attributi (e relativi tipi)

◆ Istanza del risultato

- tutte le ennuple ottenute concatenando ennuple delle due tabelle
- indiscriminatamente (in tutti i modi possibili)
- cardinalità pari al prodotto delle cardinalità

Join

◆ Prodotto cartesiano

- consente di correlare dati di tabelle diverse
- ma genera risultati di grandi dimensioni
 - es: tabelle di 1000 ennuple > 1 mil. di ennuple
- su cui poi bisogna effettuare una selezione

◆ Sarebbe opportuno

- avere un operatore che consente di fare la correlazione verificando contestualmente la condizione

“Professori e Numeri: Alternativa”

Numeri

Professori

cod	cognome	nome	qualifica	facolta
FT	Totti	Francesco	ordinario	Ingegneria
CV	Vieri	Christian	associato	Scienze
ADP	Del Piero	Alessandro	supplente	null

professore	numero
FT	0971205145
FT	347123456
CV	0971205227
ADP	0971205363
ADP	338123456

I Passo: Join

TabellaA' = Professori \bowtie cod=professore Numeri

cod	cognome	nome	qualifica	facolta	professore	numero
FT	Totti	Francesco	ordinario	Ingegneria	FT	0971205145
FT	Totti	Francesco	ordinario	Ingegneria	FT	347123456
CV	Vieri	Christian	associato	Scienze	CV	0971205227
ADP	Del Piero	Alessandro	supplente	null	ADP	0971205363
ADP	Del Piero	Alessandro	supplente	null	ADP	338123456

“Professori e Numeri”

Tabella A'

cod	cognome	nome	qualifica	facolta	professore	numero
FT	Totti	Francesco	ordinario	Ingegneria	FT	0971205145
FT	Totti	Francesco	ordinario	Ingegneria	FT	347123456
CV	Vieri	Christian	associato	Scienze	CV	0971205227
ADP	Del Piero	Alessandro	supplente	null	ADP	0971205363
ADP	Del Piero	Alessandro	supplente	null	ADP	338123456

II Passo: Proiezione ProfessoriENumeri= π cognome,nome,numero (TabellaA')

cognome	nome	numero
Totti	Francesco	0971205145
Totti	Francesco	347123456
Vieri	Christian	0971205227
Del Piero	Alessandro	0971205363
Del Piero	Alessandro	338123456

Professori e Numeri

◆ In sintesi

TabellaA' = Professori $\bowtie_{\text{cod=professore}}$ Numeri

ProfessoriENumeri = $\pi_{\text{cognome, nome, numero}}$ (TabellaA')

◆ Ovvero

ProfessoriENumeri = $\pi_{\text{cognome, nome, numero}}$ (
Professori $\bowtie_{\text{cod=professore}}$ Numeri)

Join: Sintassi

◆ Operatore binario (“diadico”)

◆ Sintassi

R $\bowtie_{\text{condizione}}$ S

◆ Condizione

- “attributo di R” = “attributo di S”
- AND di condizioni semplici
- gli attributi confrontati devono essere dello stesso tipo

Join: Semantica

◆ Schema del risultato

- unione degli attributi e relativi tipi

◆ Istanza del risultato:

- ennuple ottenute concatenando ennuple di R ed ennuple di S, tali che soddisfano la condizione

◆ Equivalente a prodotto cartesiano e selezione:

$$R \bowtie_{\text{condizione}} S = \sigma_{\text{condizione}} (R \times S) \quad \text{Attenzione: semantica "operazionale" diversa}$$

Join Complessi

◆ Esempio

- “Matricola e cognome degli studenti che hanno sostenuto l'esame di informatica teorica”

◆ Richiede di correlare tre tabelle

- matricola di Studenti e studente di Esami
- cod di Corso e corso di Esami

◆ Due condizioni di join

“Esame di Informatica Teorica”

```
TABLE Studenti (  
  matr integer PRIMARY KEY,  
  cognome varchar(20) NOT NULL,  
  nome varchar(20) NOT NULL,  
  ciclo char(20),  
  anno integer,  
  relatore char(4)  
  REFERENCES Professori(cod));  
TABLE Esami (  
  studente integer  
  REFERENCES Studenti(matr)  
  corso char(3)  
  REFERENCES Corsi(cod),  
  voto integer,  
  lode bool,  
  PRIMARY KEY (studente, corso));
```

StudentiEsami = Studenti  matr=studente Esami


```
TABLE StudentiEsami (  
  matr integer,  
  cognome varchar(20),  
  nome varchar(20),  
  ciclo char(20),  
  anno integer,  
  relatore char(4)  
  studente integer,  
  corso char(3),  
  voto integer,  
  lode bool);
```

“Esame di Informatica Teorica”

```
TABLE StudentiEsami (  
  matr integer,  
  cognome varchar(20),  
  nome varchar(20),  
  Studenti.ciclo char(20),  
  anno integer,  
  relatore char(4)  
  studente integer,  
  corso char(3),  
  voto integer,  
  lode bool);  
TABLE Corsi (  
  cod char(3) PRIMARY KEY,  
  titolo varchar(20) NOT NULL,  
  ciclo char(20),  
  docente char(4)  
  REFERENCES Professori(cod));
```

```
TABLE StudentiEsamiCorsi (  
  matr integer,  
  cognome varchar(20),  
  nome varchar(20),  
  Studenti.ciclo char(20),  
  anno integer,  
  relatore char(4)  
  studente integer,  
  corso char(3),  
  voto integer,  
  lode bool  
  cod char(3),  
  titolo varchar(20),  
  Corso.ciclo char(20),  
  docente char(4));
```

StudentiEsamiCorsi =
StudentiEsami

 corso=cod
Corsi

“Esame di Informatica Teorica”

matr	cognome	nome	...	studente	corso	voto	...	cod	titolo	...
111	Rossi	Mario	...	111	PR1	27	...	PR1	Progr...	...
222	Neri	Paolo	...	222	ASD	30	...	ASD	Algorit...	...
111	Rossi	Mario	...	111	INFT	24	...	INFT	Inform...	...
77777	Bruno	Pasquale	...	77777	PR1	21	...	PR1	Progr...	...
77777	Bruno	Pasquale	...	77777	ASD	20	...	ASD	Algorit...	...
88888	Pinco	Pietro	...	88888	ASD	28	...	ASD	Algorit...	...
88888	Pinco	Pietro	...	88888	PR1	30	...	PR1	Progr...	...
88888	Pinco	Pietro	...	88888	INFT	30	...	INFT	Inform...	...

dalla tabella Studenti
dalla tabella Esami
dalla tabella Corsi

Risultato = π matr, cognome (σ titolo='Inform. t.' (StudentiEsamiCorsi))

Risultato

matr	cognome
111	Rossi
88888	Pinco

Join Complessi

StudentiEsami = Studenti \bowtie matr=studente Esami

StudentiEsamiCorsi = StudentiEsami \bowtie cod=corso Corsi

StudentiEsamiCorsi =

(Studenti \bowtie matr=studente Esami) \bowtie cod=corso Corsi

Il join è associativo:

StudentiEsamiCorsi =

Studenti \bowtie matr=studente (Esami \bowtie cod=corso Corsi)

StudentiEsamiCorsi =

Studenti \bowtie matr=studente Esami \bowtie cod=corso Corsi

“Esame di Informatica Teorica”

◆ In sintesi:

StudentiEsami = Studenti $\bowtie_{\text{matr=studente}}$ Esami

StudentiEsamiCorsi = StudentiEsami $\bowtie_{\text{cod=corso}}$ Corsi

Risultato = $\pi_{\text{matr, cognome}}$ (

$\sigma_{\text{titolo='Inform. t.'}}$ (StudentiEsamiCorsi))

Risultato = $\pi_{\text{matricola, cognome}}$ ($\sigma_{\text{titolo='Inform. t.'}}$ (

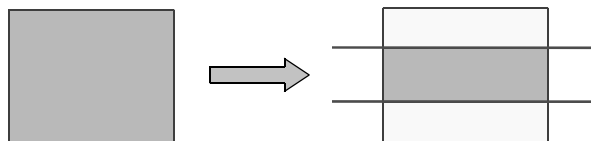
Studenti $\bowtie_{\text{matr=studente}}$ Esami $\bowtie_{\text{cod=corso}}$ Corsi))

Algebra Relazionale

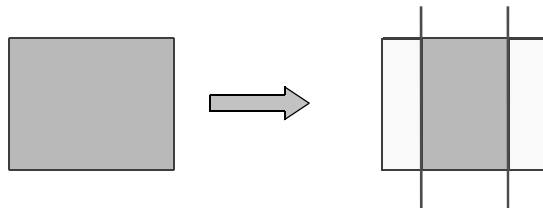
Operatori fondamentali visti fin qui:

◆ Unari

- selezione, $\sigma_{\text{condizione}}(R)$



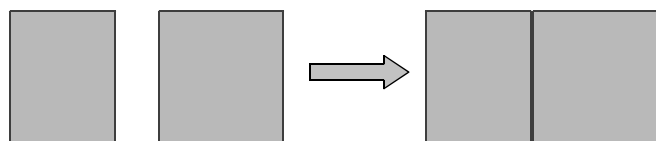
- proiezione, $\pi_{\text{attributi}}(R)$



◆ Binari

- prodotto cartesiano, $R \times S$

- join, $R \bowtie_{\text{condizione}} S$



Algebra Relazionale

◆ Altri operatori importanti

- Operatori insiemistici
 - unione
 - intersezione
 - differenza
- Ridenominazione
 - meno importante (ha solo funzioni di leggibilità)

Operatori Insiemistici

◆ Funzione

- le tabelle sono collezioni (=insiemi) di ennuple
- è possibile applicare le operazioni consuete sulle collezioni

◆ Operazioni di base

- unione
- intersezione
- differenza

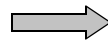
Operatori Insiemistici

◆ Esempio

- “Cognome e nome di tutte le persone”

```
TABLE Professori (  
  cod char(4) PRIMARY KEY,  
  Professori.cognome varchar(20) NOT NULL,  
  Professori.nome varchar(20) NOT NULL,  
  qualifica char(15),  
  facolta char(10) );
```

```
TABLE Studenti (  
  matr integer PRIMARY KEY,  
  Studenti.cognome varchar(20) NOT NULL,  
  Studenti.nome varchar(20) NOT NULL,  
  ciclo char(20),  
  anno integer,  
  relatore char(4)  
  REFERENCES Professori (cod) );
```



```
TABLE Persone (  
  cognome varchar(20),  
  nome varchar(20));
```

“Cognome e Nome delle Persone”

Professori

NomiProfessori = $\pi_{\text{cognome, nome}}$ (Professori)

cod	cognome	nome	qualifica	facolta
FT	Totti	Francesco	ordinario	Ingegneria
CV	Vieri	Christian	associato	Scienze
ADP	Del Piero	Alessandro	supplente	null

cognome	nome
Totti	Francesco
Vieri	Christian
Del Piero	Alessandro

Studenti

NomiStudenti = $\pi_{\text{cognome, nome}}$ (Studenti)

matr	cognome	nome	ciclo	anno	relatore
111	Rossi	Mario	laurea tr.	1	null
222	Neri	Paolo	laurea tr.	2	null
333	Rossi	Maria	laurea tr.	1	null
444	Pinco	Palla	laurea tr.	3	FT
77777	Bruno	Pasquale	laurea sp.	1	FT
88888	Pinco	Pietro	laurea sp.	1	CV

cognome	nome
Rossi	Mario
Neri	Paolo
Rossi	Maria
Pinco	Palla
Bruno	Pasquale
Pinco	Pietro

“Cognome e Nome delle Persone”

NomiProfessori

cognome	nome
Totti	Francesco
Vieri	Christian
Del Piero	Alessandro

Persone =

NomiProfessori \cup NomiStudenti

NomiStudenti

cognome	nome
Rossi	Mario
Neri	Paolo
Rossi	Maria
Pinco	Palla
Bruno	Pasquale
Pinco	Pietro

cognome	nome
Totti	Francesco
Vieri	Christian
Del Piero	Alessandro
Rossi	Mario
Neri	Paolo
Rossi	Maria
Pinco	Palla
Bruno	Pasquale
Pinco	Pietro

Operatori Insiemistici: Sintassi

◆ Operatori binari

◆ Sintassi

- stessi simboli delle operazioni convenzionali su insiemi

◆ Unione: $R \cup S$

◆ Intersezione: $R \cap S$

◆ Differenza: $R - S$

Operatori Insiemistici: Semantica

◆ Si applicano solo in alcuni casi

- le tabelle R ed S devono avere lo stesso numero di attributi
- associazione posizionale: gli attributi devono avere ordinatamente lo stesso tipo
 - N.B.: i nomi degli attributi possono essere anche diversi!

◆ Schema del risultato

- eredita i nomi degli attributi dalla prima tabella

Operatori Insiemistici: Semantica

◆ Istanza del risultato

- unione, intersezione o differenza delle ennuple

◆ Attenzione

- semantica della differenza: “tutti gli elementi del primo membro che non appartengono al secondo”
- esempio: $\{1, 3, 5\} - \{3, 7, 9\} = \{1, 5\}$

Operatori Insiemistici: Semantica

◆ Attenzione

- dal risultato degli operatori insiemistici vengono eliminati eventuali duplicati
- passo finale di eliminazione degli eventuali duplicati prodotti

◆ Motivazioni

- semantica più naturale
- esistono implementazioni efficienti

Operatori Insiemistici: Semantica

◆ Esempio

NomiProfessori

cognome	nome
Pinco	Palla
Vieri	Christian
Del Piero	Alessandro

NomiStudenti

cognome	nome
Rossi	Mario
Neri	Paolo
Rossi	Maria
Pinco	Palla
Bruno	Pasquale
Pinco	Pietro

Persone =

NomiProfessori
U
NomiStudenti

cognome	nome
Pinco	Palla
Vieri	Christian
Del Piero	Alessandro
Rossi	Mario
Neri	Paolo
Rossi	Maria
Bruno	Pasquale
Pinco	Pietro

“Cognome e Nome delle Persone”

◆ In sintesi:

NomiProfessori = $\pi_{\text{cognome, nome}}$ (Professori)

NomiStudenti = $\pi_{\text{cognome, nome}}$ (Studenti)

Persone = NomiProfessori U NomiStudenti

◆ Ovvero:

**Persone = $\pi_{\text{cognome, nome}}$ (Professori) U
 $\pi_{\text{cognome, nome}}$ (Studenti)**

“Cognome e Nome delle Persone”

◆ Nota

- sulla base della semantica è possibile anche

StranaTabella =

$\pi_{\text{nome, cognome}}$ (Professori)
U $\pi_{\text{cognome, nome}}$ (Studenti)

nome	cognome
Francesco	Totti
Christian	Vieri
Alex	Del Piero
Rossi	Mario
Neri	Paolo
Rossi	Maria
Pinco	Palla
Bruno	Pasquale
Pinco	Pietro

Intersezione / differenza

- ◆ **Elenco degli studenti che hanno omonimi tra i professori**
 - Intersezione (Studenti vs. Professori)

- ◆ **Elenco degli studenti che risultano iscritti a Lettere ed anche a Informatica**
 - Intersezione (Tabella iscritti a Lettere, tab. iscritti Informatica)

- ◆ **Elenco dei professori che non hanno omonimi tra gli studenti**
 - Differenza (Professori vs. Studenti)

Ridenominazione

- ◆ **Funzione**
 - consente di cambiare i nomi degli attributi in una tabella temporanea
 - es: `cognomePersona`, `nomePersona`

- ◆ **Caratteristiche**
 - agisce solo sullo schema
 - non cambia né la cardinalità, né il n. di attributi
 - normalmente si applica solo sul risultato finale (funzione puramente “cosmetica”)

Ridenominazione

◆ Sintassi

- $\rho_{ridenominazioni}$ (R)
- ridenominazioni: elenco di coppie *vecchioNome AS nuovoNome* separate da virgole
- *vecchioNome* deve essere un attributo di R

◆ Semantica

- nello schema del risultato ai vecchi nomi sono sostituiti i nuovi

“Cognomi e Nomi delle Persone”

◆ In definitiva:

Persone =

$\rho_{\text{cognome AS cognomePersona, nome AS nomePersona}}$ ($\pi_{\text{cognome, nome}}$ (Professori))
 \cup $\pi_{\text{cognome, nome}}$ (Studenti)

```
TABLE Persone (  
    cognomePersona varchar(20) ,  
    nomePersona varchar(20) );
```

Forma Standard

◆ Interrogazioni in algebra relazionale

- risultato dell'applicazione di vari operatori
- è possibile applicare gli operatori in ordine vario (es: prima selezioni o prima ridenominazioni)

◆ Forma standard

- nel seguito viene presentata una strategia sistematica per la scrittura di interrogazioni
- ordine standardizzato di applicazione degli operatori dell'algebra

Forma Standard

◆ Esempio

- “Nome e Cognome dei professori ordinari che non hanno tesisti della laurea triennale”

◆ Strategia

- (a) trovo nome e cognome di tutti i professori ordinari
- (b) trovo nome e cognome dei professori che hanno tesisti della laurea triennale
- faccio la differenza tra (a) e (b)

(a) “Cognomi e Nomi dei Prof. Ordinari”

Professori

<u>cod</u>	cognome	nome	qualifica	facolta
FT	Totti	Francesco	ordinario	Ingegneria
CV	Vieri	Christian	associato	Scienze
ADP	Del Piero	Alessandro	supplente	null

ProfessoriOrdinari = $\sigma_{\text{qualifica} = \text{'Ordinario'}}(\text{Professori})$

<u>cod</u>	cognome	nome	qualifica	facolta
FT	Totti	Francesco	ordinario	Ingegneria

NomiOrdinari = $\rho_{\text{cognome AS cognomeProf, nome AS nomeProf}}(\pi_{\text{cognome, nome}}(\text{ProfessoriOrdinari}))$

cognomeProf	nomeProf
Totti	Francesco

(b) “Cognomi e Nomi di Prof. con Tesi Triennali”

Professori

<u>cod</u>	cognome	nome	qualifica	facolta
FT	Totti	Francesco	ordinario	Ingegneria
CV	Vieri	Christian	associato	Scienze
ADP	Del Piero	Alessandro	supplente	null

Studenti

<u>matr</u>	cognome	nome	ciclo	anno	relatore
111	Rossi	Mario	laurea tr.	1	null
222	Neri	Paolo	laurea tr.	2	null
333	Rossi	Maria	laurea tr.	1	null
444	Pinco	Palla	laurea tr.	3	FT
77777	Bruno	Pasquale	laurea sp.	1	FT
88888	Pinco	Pietro	laurea sp.	1	CV

ProfessoriConTesiisti = Studenti $\bowtie_{\text{relatore} = \text{cod}}$ Professori

(b) “Cognomi e Nomi di Prof. con Tesi Triennali”

ProfessoriConTesisti = Studenti \bowtie relatore = cod Professori

matr	S.cognome	S.nome	ciclo	anno	relatore	P.cognome	P.nome	qualifica	facolta
444	Pinco	Palla	laurea tr.	3	FT	Totti	Francesco	ordinario	Ing.
77777	Bruno	Pasquale	laurea sp.	1	FT	Totti	Francesco	ordinario	Ing.
88888	Pinco	Pietro	laurea sp.	1	CV	Vieri	Christian	associato	Sc.

ProfessoriConTesiTriennali = $\sigma_{\text{ciclo} = \text{'laurea tr.'}}$ (ProfessoriConTesisti)

matr	S.cognome	S.nome	ciclo	anno	relatore	P.cognome	P.nome	qualifica	facolta
444	Pinco	Palla	laurea tr.	3	FT	Totti	Francesco	ordinario	Ing.

NomiProfConTesiTriennali = $\rho_{\text{cognome AS cognomeProf, nome AS nomeProf}}$ ($\pi_{\text{Professori.cognome, Professori.nome}}$ (ProfessoriConTesiTriennali))

cognomeProf	nomeProf
Totti	Francesco

“Cognomi e Nomi dei Prof. Ordinari senza Tesi Triennali”

NomiOrdinari =

$\rho_{\text{cognome AS cognomeProf, nome AS nomeProf}}$ ($\pi_{\text{cognome, nome}}$ ($\sigma_{\text{qualifica} = \text{'Ordinario'}}$ (Professori)))

cognomeProf	nomeProf
Totti	Francesco

NomiProfConTesiTriennali =

$\rho_{\text{cognome AS cognomeProf, nome AS nomeProf}}$ ($\pi_{\text{cognome, nome}}$ ($\sigma_{\text{ciclo} = \text{'laurea tr.'}}$ (Studenti \bowtie relatore = cod Professori))))

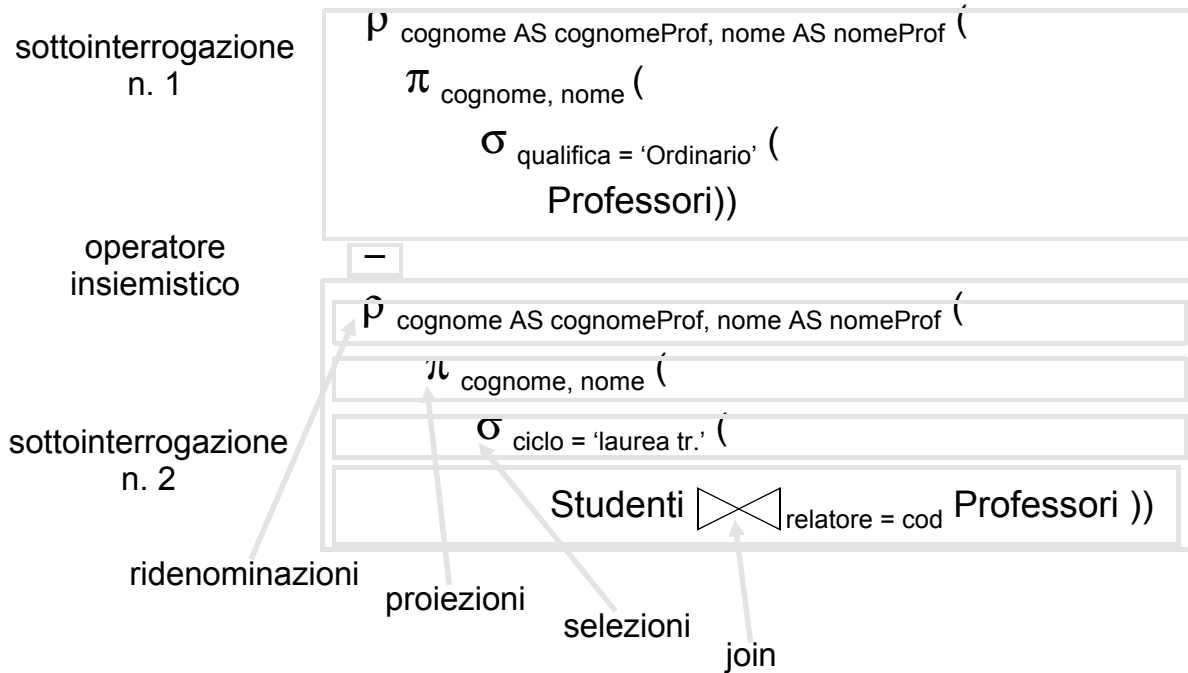
cognomeProf	nomeProf
Totti	Francesco

Risultato = NomiOrdinari – NomiProfConTesiTriennali

cognomeProf	nomeProf

Riassumendo

Risultato =



Forma Standard

- ◆ **Varie sottointerrogazioni (una o più)**
 - correlate con operatori insiemistici
- ◆ **Ciascuna sottointerrogazione**
 - prima: eventuali join o prodotti cartesiani
 - poi: eventuali selezioni
 - poi: eventuali proiezioni
 - infine: eventuali ridenominazioni