



Informatica **U**manistica

# Basi di Dati

*Algebra Relazionale*

*Dettagli e Approfondimenti*



UNIVERSITÀ DI PISA

## Dettagli e Approfondimenti

- ◆ **Selezioni e Valori Nulli**
- ◆ **Proiezione e Duplicati**
- ◆ **Altre Forme di Join**
- ◆ **Altri Operatori**
  - Ordinamento
  - Funzioni Aggregative
  - Alias
- ◆ **Forma Standard Completa**

# Algebra Relazionale

- ◆ Linguaggio alla base del DML di SQL
- ◆ Basato sull'algebra teorica di Codd
- ◆ Insieme di operatori su tabelle
- ◆ che producono tabelle
- ◆ Espressioni complesse
- ◆ Assegnazioni

# Selezione

## ◆ Selezione

- seleziona alcune tra le ennuple di una tabella
- condizione sui valori degli attributi

## ◆ Sintassi

- $\sigma_{\text{condizione}}(R)$ ; es:  $\sigma_{\text{ciclo}='laurea\ tr.' \text{ OR } \text{anno}=2}(\text{Studenti})$

## ◆ Semantica

- seleziona nel risultato le ennuple di R che soddisfano la selezione

## Selezione e Valori Nulli

### ◆ **Attenzione ai valori nulli**

- le condizioni sono sempre false in presenza di valori nulli
- es: facolta<>'Ingegneria' solo le ennuple per cui la facoltà è non nulla e diversa da ing.
- condizioni speciali: IS NULL, IS NOT NULL
- es: facolta<>'Ingegneria' OR facolta IS NULL tutte le ennuple in cui il valore non è ing.

## “Professori che Non Sono di Ingegneria”

Professori

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

ProfessoriAltreFacolta =  $\sigma_{\text{facolta} \neq \text{'Ingegneria'}}(\text{Professori})$

<u>cod</u>	cognome	nome	qualifica	facolta
CV	Vieri	Christian	associato	Scienze

ProfessoriNoIngegneria =  $\sigma_{\text{facolta} \neq \text{'Ingegneria'} \text{ OR } \text{facolta IS NULL}}(\text{Professori})$

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

# Proiezione

## ◆ Proiezione

- elimina alcuni tra gli attributi di una tabella
- lista attributi da mantenere

## ◆ Sintassi

- $\pi_{\text{attributi}}(R)$ ; es:  $\pi_{\text{cognome, nome, ciclo}}(\text{Studenti})$

## ◆ Semantica

- genera una tabella che contiene la restrizione delle ennuple originali agli attributi specificati

## Proiezione e Duplicati

- ◆ **Può generare duplicati**
  - nel caso nel risultato non sopravvivano chiavi
  
- ◆ **Filosofia dell'SQL (e quindi dell'algebra)**
  - l'utente può scegliere se eliminare o meno i duplicati
  
- ◆ **Operatore DISTINCT**
  - sintassi: `DISTINCT (R)`
  - semantica: elimina da R i duplicati



## “Cognomi e Anni di Corso 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

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

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

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

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

## Operatori Binari

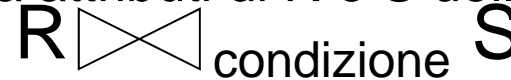
### ◆ Prodotto Cartesiano

- operatore binario  $R \times S$
- correla le ennuple di  $R$  con quelle di  $S$  in tutti i possibili modi
- il risultato ha tutti gli attributi di  $S$  e tutti gli attributi di  $R$
- $|R \times S| = |R| * |S|$
- prestazioni scadenti

## Operatori Binari

### ◆ Join

- operatore binario:
- condizione: uguaglianza tra attributi di R e S dello stesso tipo; es:  $R.a = S.b$
- il risultato è l'insieme di ennuple ottenute "concatenando" le ennuple di R con quelle di S, purché l'ennupla risultante soddisfi la condizione
- consente implementazioni efficienti



## Operatori Binari

### ◆ Nota

- le condizioni di join tra le tabelle corrispondono normalmente ai vincoli di riferimento (es: studente ed esame)
- ma non necessariamente è sempre così
- es: join tra Professori e Studenti su Professori.nome=Studenti.nome
- es: join tra Studenti e Corsi su Studenti.cognome=Corsi.titolo

## Altre Forme di Join

- ◆ **Ci sono numerose varianti del Join**
- ◆ **Join Naturale**
  - variante del join senza condizione
  - condizione implicita: valori uguali sugli attributi con lo stesso nome
- ◆ **Poco utilizzato**
- ◆ **Più utilizzati sono i join esterni**

## Cardinalità del Join

- ◆ **In un join ci possono essere ennuple di R e di S che non contribuiscono al risultato**
- ◆ **Join Completo**
  - tutte le ennuple di R e le ennuple di S contribuiscono ad almeno un'ennupla del risultato
- ◆ **Join Non Completo**
  - non tutte le ennuple partecipano al risultato
  - Join Vuoto: caso degenero

# Join Completo

Professori

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

Numeri

<u>professore</u>	<u>numero</u>
FT	0971205145
FT	347123456
VC	0971205227
ADP	0971205363
ADP	338123456

Professori  cod=professore Numeri

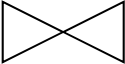
cod	cognome	nome	qualifica	facolta	professore	numero
FT	Totti	Francesco	ordinario	Ingegneria	FT	0971205145
FT	Totti	Francesco	ordinario	Ingegneria	FT	347123456
VC	Vieri	Christian	associato	Scienze	VC	0971205227
ADP	Del Piero	Alessandro	supplente	null	ADP	0971205363
ADP	Del Piero	Alessandro	supplente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

## Join Completo e Card. StudentiLaureaTriennale

CorsiLaureaTriennale

<u>cod</u>	titolo	ciclo	...
PR1	Programmazione I	I. tr.	...
ASD	Algoritmi e Str. Dati	I. tr.	...

<u>matr</u>	cognome	nome	ciclo	...
111	Rossi	Mario	I. tr.	...
222	Neri	Paolo	I. tr.	...
333	Rossi	Maria	I. tr.	...
444	Pinco	Palla	I. tr.	...

CorsiLaureaTriennale  C.ciclo = S.ciclo StudentiLaureaTriennale

<u>cod</u>	titolo	C.ciclo	...	<u>matr</u>	cognome	nome	S.ciclo	...
PR1	Programmazione I	I. tr.	...	111	Rossi	Mario	I. tr.	...
PR1	Programmazione I	I. tr.	...	222	Neri	Paolo	I. tr.	...
PR1	Programmazione I	I. tr.	...	333	Rossi	Maria	I. tr.	...
PR1	Programmazione I	I. tr.	...	444	Pinco	Palla	I. tr.	...
ASD	Algoritmi e Str. Dati	I. tr.	...	111	Rossi	Mario	I. tr.	...
ASD	Algoritmi e Str. Dati	I. tr.	...	222	Neri	Paolo	I. tr.	...
ASD	Algoritmi e Str. Dati	I. tr.	...	333	Rossi	Maria	I. tr.	...
ASD	Algoritmi e Str. Dati	I. tr.	...	444	Pinco	Palla	I. tr.	...



# Join Non Completo

Professori

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

Numeri

<u>professore</u>	<u>numero</u>
FT	0971205145
FT	347123456
VC	0971205227
ADP	0971205363
ADP	338123456

Professori  cod=professore Numeri

supponendo che la  
ennupla sia eliminata

cod	cognome	nome	qualifica	facolta	professore	numero
FT	Totti	Francesco	ordinario	Ingegneria	FT	0971205145
FT	Totti	Francesco	ordinario	Ingegneria	FT	347123456
ADP	Del Piero	Alessandro	supplente	null	ADP	0971205363
ADP	Del Piero	Alessandro	supplente	null	ADP	338123456

# Join Vuoto

Professori

<u>cod</u>	cognome	nome	...
FT	Totti	Francesco	...
VC	Vieri	Christian	...
ADP	Del Piero	Alessandro	...

Studenti

<u>matr</u>	cognome	nome	...
111	Rossi	Mario	...
222	Neri	Paolo	...
333	Rossi	Maria	...
444	Pinco	Palla	...
77777	Bruno	Pasquale	...
88888	Pinco	Pietro	...

Professori  Professori.nome = Studenti.nome Studenti

<u>matr</u>	S.cognome	S.nome	ciclo	anno	relatore	P.cognome	P.nome	qualifica	facolta
-------------	-----------	--------	-------	------	----------	-----------	--------	-----------	---------

## Cardinalità del Join

- ◆ **Cardinalità del join di  $R_1$  e  $R_2$** 
  - compreso fra zero e  $|R_1| * |R_2|$
- ◆ **Caso tipico:**
  - join tra la chiave primaria di  $R_1$  e un attributo di  $R_2$  su cui c'è un vincolo di integrità referenziale
  - cardinalità pari a  $|R_2|$
  - es: Prof. e Numeri, con Prof.cod=Numeri.professore
- ◆ **Se non ci fosse il vincolo di integrità**
  - cardinalità compresa fra zero e  $|R_2|$

## Join Esterni

- ◆ **Nei join incompleti si “perdono” ennuple**
  - es: professori senza numeri di telefono
- ◆ **In alcuni casi può essere un problema**
- ◆ **Join Esterno (“Outer Join”)**
  - tutte le ennuple contribuiscono (completo)
  - le ennuple per cui non c'è controparte sono completate con valori nulli

## Join Esterni

### ◆ Join Esterno Sinistro

$\Rightarrow R \left[ \begin{array}{c} \diagup \\ \diagdown \end{array} \right] \text{condizione} S$

- garantisce che vengono mantenute tutte le ennuple della tabella a sinistra (R)

### ◆ Join Esterno Destro

$\Rightarrow R \left[ \begin{array}{c} \diagdown \\ \diagup \end{array} \right] \text{condizione} S$

- garantisce che vengono mantenute tutte le ennuple della tabella a destra (S)

### ◆ Join Esterno Completo

$\Rightarrow R \left[ \begin{array}{c} \diagup \\ \diagdown \end{array} \right] \text{condizione} S$

# Join Esterno Sinistro

Professori

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

Numeri

<u>professore</u>	<u>numero</u>
FT	0971205145
FT	347123456
ADP	0971205363
ADP	338123456

Professori  cod=professore Numeri

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

# Join Esterno Compl.

Studenti

Professori

<u>cod</u>	cognome	nome	...
FT	Totti	Francesco	...
VC	Vieri	Christian	...
ADP	Del Piero	Alessandro	...

<u>matr</u>	cognome	nome	...
111	Rossi	Mario	...
222	Neri	Paolo	...
333	Rossi	Maria	...
444	Pinco	Palla	...
77777	Bruno	Pasquale	...
88888	Pinco	Pietro	...

Professori  Professori.nome = Studenti.nome Studenti

<u>cod</u>	P.cognome	P.nome	...	<u>matr</u>	S.cognome	S.nome	...
FT	Totti	Francesco	...	null	null	null	null
null	null	null	null	111	Rossi	Mario	...
...	...	...	...	...	...	...	...

## Altri Operatori

- ◆ **L'algebra offre altri operatori**
- ◆ **In particolare**
  - l'operatore ORDER BY per l'ordinamento delle ennuple in una tabella
  - le funzioni aggregative per il calcolo di valori sulla base del contenuto della base di dati
  - l'operatore ALIAS per la ridenominazione di una tabella



# Ordinamento

- ◆ **E' possibile gestire anche l'ordinamento**
  
- ◆ **Operatore ORDER BY**
  - sintassi: ORDER BY *attributi* (R)
  - *attributi*: lista di attributi di R
  - semantica: riordina le ennuple di R utilizzando i valori degli attributi specificati come chiavi di ordinamento (dal primo in avanti) in ordine crescente

## “Cognomi e Nomi degli Studenti in Ordine”

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

ORDER BY  $\pi_{\text{cognome, nome}}(\text{Studenti})$

ORDER BY  $\pi_{\text{cognome, nome}}(\text{Studenti})$

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

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

## Funzioni Aggregative

- ◆ **Finora nel risultato**
  - solo valori presenti nella base di dati iniziale
- ◆ **In alcuni casi è utile avere valori calcolati**
  - somme e medie di attributi numerici;  
es: media dei voti degli studenti
  - conteggi;  
es: numero di corsi della laurea triennale
  - massimi e minimi

# Funzioni Aggregative

## ◆ Funzione aggregativa

- argomento: attributo di una tabella
- calcolata esaminando i valori di un attributo appartenenti ad ennuple diverse

## ◆ Tipicamente

- SUM (somma), COUNT (conteggio), AVG (media), MIN (minimo), MAX (massimo)

# Funzioni Aggregative

## ◆ Sintassi

- si utilizzano nella proiezione

## ◆ Esempio:

- calcolo del voto medio degli esami
- funzione AVG() applicata all'attributo voto della tabella Esami

Risultato =  $\pi_{AVG(voto)}$  (Esami)

## Funzioni Aggregative

### ◆ Domanda

- perchè la proiezione ?
- non sarebbe più semplice:

~~Risultato = AVG(voto) (Esami)~~

### ◆ Risposta

- il risultato non può essere un numero
- deve essere una tabella (di una ennupla e una colonna)

# Funzioni Aggregative

## ◆ Semantica

- viene calcolato il risultato della proiezione sugli attributi utilizzati come argomenti
- viene applicata la funzione aggregativa ai valori dell'attributo
- il risultato è una tabella con una singola ennupla
- una colonna per ciascuna funzione aggregativa utilizzata nella proiezione

## Funzioni Aggregative

### ◆ Esempio:

**Risultato** =  $\pi_{AVG(voto)}$  (Esami)

- I passo:  $\pi_{voto}$  (Esami)
- II passo: calcolo della media dei valori dell'attributo voto
- III passo: viene creata la tabella Risultato con una unica colonna (chiamata AVG(voto)) ed un'unica ennupla, contenente il risultato



## Funzioni Aggregative: Esempi

- Voto medio degli esami

Risultato =  $\pi_{AVG(voto)}(Esami)$

$\rho_{AVG(voto) AS \text{ votomedio}}(\pi_{AVG(voto)}(Esami))$

Risultato

AVG(voto)
24,7

votomedio
24,7

- Voto massimo e minimo degli esami

$\rho_{MIN(voto) AS \text{ votomin}, MAX(voto) AS \text{ votomax}}(\pi_{MIN(voto), MAX(voto)}(Esami))$

votomin	votomax
20	30

nel caso del conteggio, l'attributo è indifferente

- Numero di corsi della laurea triennale

$\rho_{COUNT(cod) AS \text{ numcorsi}}(\pi_{COUNT(cod)}(\sigma_{ciclo='laurea tr.'}(Corsi)))$

$\rho_{COUNT(ciclo) AS \text{ numcorsi}}(\pi_{COUNT(ciclo)}(\sigma_{ciclo='laurea tr.'}(Corsi)))$

numcorsi
2

## Funzioni Aggregative

### ◆ Regola

- in una proiezione possono comparire o solo attributi ordinari, o solo funzioni aggregative
- altrimenti la semantica non è ben definita

### ◆ Esempio

- “Titolo e numero dei corsi della laurea triennale”

$\rho_{\text{COUNT(cod) AS numcorsi}} \left( \pi_{\text{titolo, COUNT(cod)}} \left( \sigma_{\text{ciclo='laurea tr.'}}(\text{Corsi}) \right) \right)$

## Alias

- ◆ **A volte la stessa tabella può essere coinvolta più di una volta nello stesso join**
- ◆ **Esempio**
  - “Cognomi e nomi degli studenti che all’esame di Programmazione hanno riportato un voto superiore a quello dei loro tutor”
- ◆ **ATTENZIONE**
  - si tratta di una interrog. molto complessa

# Alias

## ◆ In questo caso

- devo confrontare il voto dello studente nell'esame di programmazione con quello del tutor
- entrambe le informazioni vengono dalla tabella Esami (che deve necessariamente essere usata due volte)
- problema con i nomi: come distinguo l'attributo che corrisponde al voto dello studente da quello del tutor ?

# Alias

## ◆ Operatore di Alias per una Tabella

- crea una copia di una tabella esistente
- con un nome diverso (e quindi risolve il problema del nome degli attributi)

## ◆ Sintassi

- $R \text{ AS } T$

## ◆ Semantica

- l'istanza di T è identica all'istanza di R
- nello schema di T, l'attributo R.A assume il nome T.A

## Alias

- ◆ **“Cognomi e nomi degli studenti che all’esame di Programmazione hanno riportato un voto superiore a quello dei loro tutor”**
  - Studenti, per i dati degli studenti
  - Esami, per i dati sugli esami degli studenti
  - Tutorato, per le relazioni tra studenti e tutor
  - Esami di nuovo, per i dati sugli esami sostenuti dai tutor; è necessario un alias (Esami as EsamiTutor)

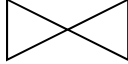
## “Studenti, Voti e Tutor”

### ◆ Strategia

- I Passo: tabella StudentiVoti, join tra Studenti ed Esami
- II Passo: tabella StudentiVotiTutor, join tra StudentiVoti e Tutorato
- III Passo: tabella StudentiVotiTutorVoti, join tra StudentiVotiTutor e Esami AS EsamiTutor

## “Studenti, Voti e Tutor”

### ◆ I Passo: join tra Studenti ed Esami

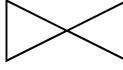
StudentiVoti= Studenti  Esami  
matr=studente

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



## “Studenti, Voti e Tutor”

### ◆ Il Passo: join con Tutorato

StudentiVotiTutor = StudentiVoti \_matr=Tutorato.studente Tutorato

```
TABLE StudentiVotiTutor (  
    Studenti.matr integer,  
    Studenti.cognome varchar(20),  
    Studenti.nome varchar(20),  
    Studenti.ciclo char(20),  
    Studenti.anno integer,  
    Studenti.relatore char(4),  
    Esami.studente integer  
    Esami.corso char(3)  
    Esami.voto integer,  
    Esami.lode bool,  
    Tutorato.studente integer,  
    Tutorato.tutor integer);
```

## “Studenti, Voti e Tutor”

### ◆ III Passo: join con Esami AS EsamiTutor

StudentiVotiTutorEsamitutor = StudentiVotiTutor

 Tutorato.tutor=EsamiTutor.studente (Esami AS EsamiTutor)

```
TABLE StudentiVotiTutorEsamitutor (  
  Studenti.matr integer,  
  Studenti.cognome varchar(20),  
  Studenti.nome varchar(20),  
  Studenti.ciclo char(20),  
  Studenti.anno integer,  
  Studenti.relatore char(4),  
  Esami.studente integer  
  Esami.corso char(3)  
  Esami.voto integer,  
  Esami.lode bool,  
  Tutorato.studente integer,  
  Tutorato.tutor integer,  
  EsamiTutor.studente integer  
  EsamiTutor.corso char(3)  
  EsamiTutor.voto integer,  
  EsamiTutor.lode bool);
```

## “Studenti, Voti e Tutor”

StudentiVotiTutorEsamitutor

Stud.ma tr	Studenti. cognome	.	Esami. studente	Esami. corso	Esami. voto	.	Tutorato. studente	Tutorato. tutor	ET. studente	ET. Corso	ET. voto	.
111	Rossi	.	111	PR1	27	.	111	77777	77777	PR1	21	.
111	Rossi	.	111	INFT	24	.	111	77777	77777	PR1	21	.
222	Neri	.	222	ASD	30	.	222	77777	77777	PR1	21	.
111	Rossi	.	111	PR1	27	.	111	77777	77777	ASD	20	.
111	Rossi	.	111	INFT	24	.	111	77777	77777	ASD	20	.
222	Neri	.	222	ASD	30	.	222	77777	77777	ASD	20	.

tutte le possibili coppie fatte di  
un esame di uno studente ed un esame del suo tutor

## “Studenti, Voti e Tutor”

### ◆ Selezioni e proiezioni finali

Risultato =  $\pi$  cognome, nome (

$\sigma$  Esami.corso='Pr1' AND EsamiTutor.corso='Pr1' AND Esami.voto > EsamiTutor.voto (

StudentiVotiTutorEsamitutor )

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

## “Studenti, Voti e Tutor”

### ◆ Riassumendo

Risultato =  $\pi$  cognome, nome (

$\sigma$  Esami.corso='Pr1' AND EsamiTutor.corso='Pr1' AND Esami.voto > EsamiTutor.voto (

Studenti  matr=studente Esami

 matr=Tutorato.studente Tutorato

 Tutorato.tutor=EsamiTutor.studente (Esami AS EsamiTutor)))

## Forma Standard Completa

- ◆ **Alla luce dei nuovi operatori, possiamo completare la forma standard**
- ◆ **Varie sottointerrogazioni (una o più)**
  - correlate con operatori insiemistici
- ◆ **Per ogni sottointerrogazione**
  - una o più tabelle
  - eventuali alias
  - eventuali join o prodotti cartesiani



## Forma Standard Completa (Continua)

### ◆ Per ogni sottointerrogazione (continua)

- eventuali selezioni
- eventuali proiezioni, con eventuali funzioni aggregative
- eventuale eliminazione dei duplicati
- eventuali ridenominazioni
- eventuali riordinamenti

## Metodo di Scrittura Completo

- ◆ **Stabilire se sono necessari operatori insiemistici**
  - dividere in sottointerrogazioni
  
- ◆ **Per ogni sottointerrogazione**
  - decidere da quali tabelle prelevare i dati
  
- ◆ **Decidere eventuali alias**
  
- ◆ **Se le tabelle sono più di una,**
  - strategia a) prodotti cartesiani oppure
  - strategia b) join con le condizioni opportune



## Metodo di Scrittura Completo (Cont.)

- ◆ **Scrivere le eventuali selezioni**
  - strategia a) incluse le condizioni di Join
- ◆ **Scrivere le eventuali proiezioni**
  - e le eventuali funzioni aggregative
- ◆ **Scrivere le eventuali eliminazione di duplicati**
- ◆ **Scrivere le eventuali ridenominazioni finali**
- ◆ **Scrivere gli eventuali operatori di ordinamento**
- ◆ **Rimettere le sottointerrogazioni insieme**

## Metodo di Scrittura Completo

### ◆ Suggestimento

- dividere la scrittura dell'interrogazione in passi, producendo ad ogni passo un risultato intermedio
- ragionare sulla struttura del risultato intermedio (schema, ovvero attributi, e istanza, ovvero numero e natura delle ennuple)

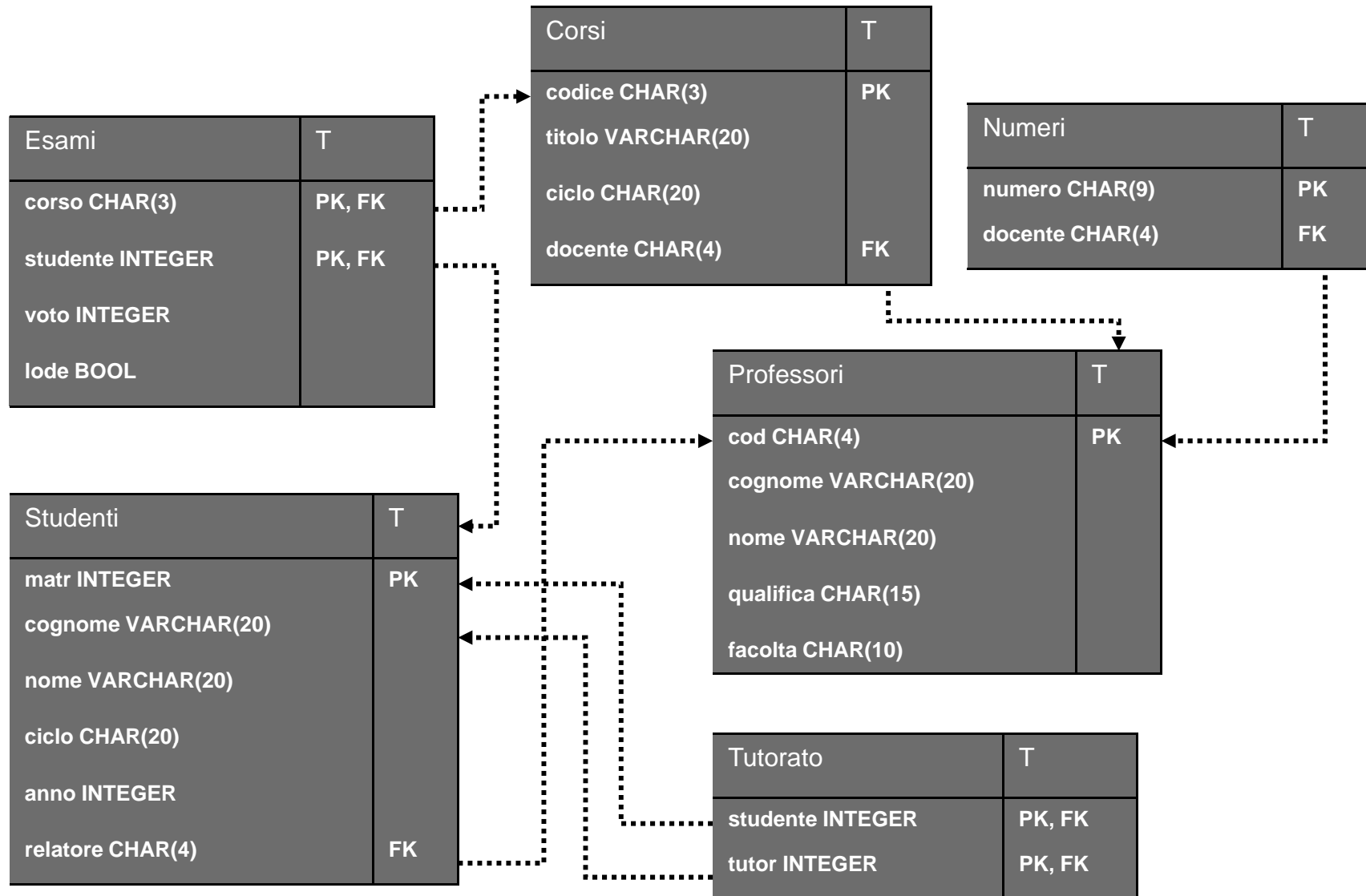
## Dettagli e Approfondimenti

- ◆ **Selezioni e Valori Nulli**
- ◆ **Proiezione e Duplicati**
- ◆ **Altre Forme di Join**
- ◆ **Altri Operatori**
  - Ordinamento
  - Funzioni Aggregative
  - Alias
- ◆ **Forma Standard Completa**

## Algebra Relazionale >> La Base di Dati di Esempio

```
TABLE Professori (  
    cod char(4) PRIMARY KEY,  
    cognome varchar(20) NOT NULL,  
    nome varchar(20) NOT NULL,  
    qualifica char(15),  
    facolta char(10) );  
  
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 Corsi (  
    cod char(3) PRIMARY KEY,  
    titolo varchar(20) NOT NULL,  
    ciclo char(20),  
    docente char(4)  
        REFERENCES Professori(cod)  
);  
  
TABLE Tutorato (  
    studente integer  
        REFERENCES Studenti(matr),  
    tutor integer  
        REFERENCES Studenti(matr),  
    PRIMARY KEY (studente,tutor));  
  
TABLE Esami (  
    studente integer  
        REFERENCES Studenti(matr)  
    ON DELETE cascade  
    ON UPDATE cascade,  
    corso char(3)  
        REFERENCES Corsi(cod),  
    voto integer,  
    lode bool,  
    CHECK (voto>=18 and voto<=30),  
    CHECK (not lode or voto=30),  
    PRIMARY KEY (studente, corso));  
  
TABLE Numeri (  
    professore char(4)  
        REFERENCES Professori(cod),  
    numero char(9),  
    PRIMARY KEY (professore,numero));
```

## Algebra Relazionale >> La Base di Dati di Esempio



## Algebra Relazionale >> La Base di Dati di Esempio

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

## Algebra Relazionale >> La Base di Dati di Esempio

Tutorato

<u>studente</u>	<u>tutor</u>
111	77777
222	77777
333	88888
444	88888

Numeri

<u>professore</u>	<u>numero</u>
FT	0971205145
FT	347123456
VC	0971205227
ADP	0971205363
ADP	338123456

Esami

<u>studente</u>	<u>corso</u>	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