

Informatica**Umanistica**

Basi di Dati

Algebra Relazionale

Concetti Fondamentali



UNIVERSITÀ DI PISA

Concetti Fondamentali

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

Introduzione

- ◆ **SQL – Structured Query Language**
- ◆ **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 ennuple
 - 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; analogo:
- semantica operativa astratta
analogo: somme per addizioni di unità

$$W = \frac{3x}{y-3}$$

Introduzione

◆ Algebra Relazionale

- definita da Codd nell'articolo sul modello
- lontana dal'SQL sotto molti aspetti

◆ Algebra dei DBMS Relazionali

- operatori attraverso cui vengono implementate le interrogazioni SQL
- alla base delle ottimizzazioni

◆ Presenteremo la seconda

Base di Dati dei Corsi di Informatica

◆ Professori

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

◆ Studenti

- matricola, nome, tipo di corso (ciclo): laurea tr., laurea spec.
- 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 char(20) NOT NULL,
    nome char(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));

```

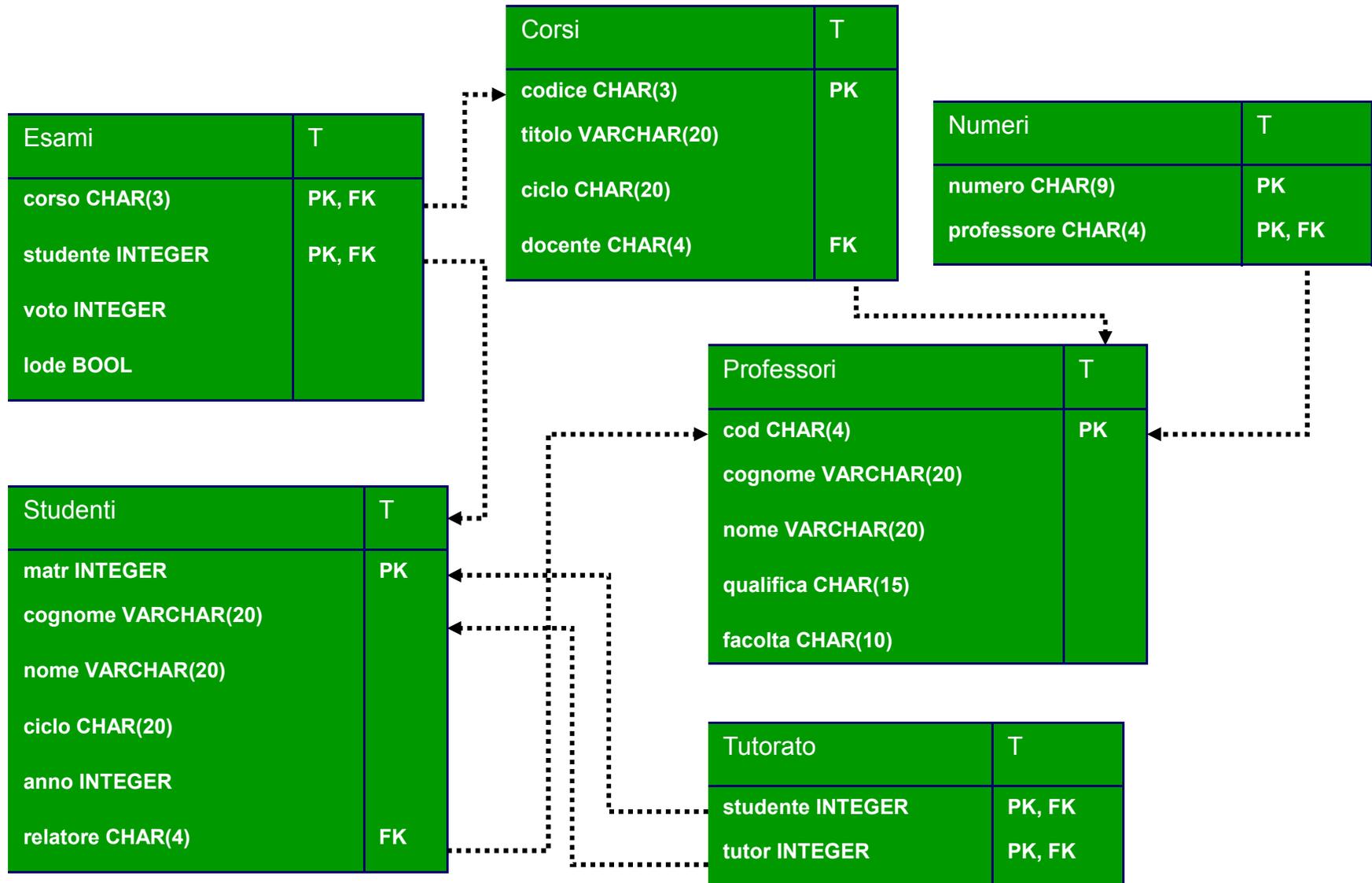
```

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

```

Base di Dati dei Corsi di Informatica

- ◆ **Una rappresentazione grafica dello schema relazionale (stereotipi di UML)**
 - ogni tabella è un rettangolo con un nome e la lista degli attributi
 - le chiavi primarie sono indicate con PK
 - le chiavi esterne sono indicate con FK
 - i vincoli di riferimento sono indicati da archi tra le tabelle (utile per visualizzarli immediatamente)



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

<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>	<u>voto</u>	<u>lode</u>
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

Base di Dati dei Corsi di Informatica

◆ Nota

- nella base di dati vengono utilizzati ripetutamente codici alfanumerici
- leggibilità dell'esempio (es: 'FT', 'PR1')
- in generale è meglio adottare come chiavi primarie codici interi
- maggiore compattezza e migliori prestazioni

Algebra Relazionale

◆ Collezione di operatori

- applicati a tabelle
- che producono 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

◆ **Tabelle originali della base di dati**

- con schema completo di vincoli

◆ **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

◆ Terminologia

- ispirata all'analogia ennuple = vettori

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:**

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

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

“Studenti della laurea tr. di anni successivi al I”

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
7777	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 = $\sigma_{\text{ciclo='laurea tr.' AND anno>1}}$ (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 (  
  Studenti.matr integer,  
  Studenti.cognome varchar(20),  
  Studenti.nome varchar(20),  
  Studenti.ciclo char(20),  
  Studenti.anno integer,  
  Studenti.relatore char(4));
```

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

Proiezione

◆ Funzione

- estrarre alcune delle colonne di una tabella

◆ Esempio

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

```
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	FI
7777	Bruno	Pasquale	laurea sp.	1	FI
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

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

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

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

<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

Risultato

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

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

ennuple identiche
(duplicati)

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

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

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

I Passo: Prodotto Cartesiano

TabellaA = Professori X Numeri

<u>cod</u>	<u>cognome</u>	<u>nome</u>	<u>qualifica</u>	<u>facolta</u>	<u>professore</u>	<u>numero</u>
FT	Totti	Francesco	ordinario	Ingegneria	FT	0971205145
FT	Totti	Francesco	ordinario	Ingegneria	FT	347123456
FT	Totti	Francesco	ordinario	Ingegneria	VC	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	VC	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

Il Passo: Selezione $TabellaB = \sigma_{cod=professore}(TabellaA)$

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	VC	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

III Passo: Proiezione ProfessoriENumeri = π cognome,nome,numero (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)

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

Prodotto Cartesiano: Sintassi

- ◆ **Operatore binario (“diadico”)**
 - è il primo operatore che consente di mettere assieme dati provenienti da tabelle diverse
- ◆ **Sintassi**
 - $R \times S$

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

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

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

I Passo: Join

TabellaA' = 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

“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	VC	0971205227
ADP	Del Piero	Alessandro	supplente	null	ADP	0971205363
ADP	Del Piero	Alessandro	supplente	null	ADP	338123456

Il Passo: Proiezione ProfessoriENumeri = $\pi_{\text{cognome, nome, numero}}$ (Tabella A')

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 _{cod=professore} Numeri

ProfessoriENumeri = π _{cognome, nome, numero} (TabellaA')

ProfessoriENumeri = π _{cognome, nome, numero} (
Professori \bowtie _{cod=professore} Numeri)

Join: Sintassi

- ◆ Operatore binario (“diadico”)
- ◆ Sintassi

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

- ◆ **Condizione**
 - AND di condizioni semplici
 - attributo di R = attributo di S
 - i due attributi 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:

$$R \bowtie_{\text{condizione}} S = \sigma_{\text{condizione}} (R \times S)$$

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  Esami
matr=studente

```
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);
```

StudentiEsamiCorsi =

StudentiEsami



corso=cod

Corso

```
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));
```

“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 matricola, 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 Infomatica Teorica”

◆ In sintesi:

StudentiEsami = Studenti \bowtie _{matr=studente} Esami

StudentiEsamiCorsi = StudentiEsami \bowtie _{cod=corso} Corsi

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

Risultato = π matricola, cognome (σ titolo='Inform. t.' (
Studenti \bowtie _{matr=studente} Esami \bowtie _{cod=corso} Corsi))

Algebra Relazionale

◆ Finora

- abbiamo visto gli operatori fondamentali

◆ 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 di ennuple
- è possibile applicare le operazioni consuete sulle collezioni
- 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)

<u>cod</u>	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)

<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

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

NomiStudenti

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

Persone =

NomiProfessori U NomiStudenti

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
- ◆ **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

**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)
 $\cup \pi_{\text{cognome, nome}}$ (Studenti)

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

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 =

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

```
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 ridenominaz.)

◆ 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

ProfessoriConTesisti = Studenti  relatore = cod Professori

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

ProfessoriConTesisti = Studenti \bowtie relatore = cod Professori

<u>matr</u>	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)

<u>matr</u>	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 =

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

cognomeProf	nomeProf
Totti	Francesco

NomiProfConTesiTriennali =

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

cognomeProf	nomeProf
Totti	Francesco

Risultato = NomiOrdinari – NomiProfConTesiTriennali

cognomeProf

nomeProf

Riassumendo

Risultato =

sottointerrogazione
n. 1

ρ cognome AS cognomeProf, nome AS nomeProf (
 π cognome, nome (
 σ qualifica = 'Ordinario' (
Professori)))

operatore
insiemistico

—

sottointerrogazione
n. 2

ρ cognome AS cognomeProf, nome AS nomeProf (
 π cognome, nome (
 σ ciclo = 'laurea tr.' (
Studenti \bowtie relatore = cod Professori)))

ridenominazioni

proiezioni

selezioni

join

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

Forma Standard

◆ Attenzione

- la forma standard non è necessariamente la più efficiente
- esempio: anticipare la selezioni

NomiProfConTesiTriennali =

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

Forma Standard

◆ Ordine alternativo:

NomiProfConTesiTriennali =

ρ cognome AS cognomeProf, nome AS nomeProf (
 π cognome, nome (

σ ciclo = 'laurea tr.' (Studenti)  relatore = cod Professori))

esecuzione più efficiente:
si eliminano subito le ennuple
irrilevanti

Forma Standard

◆ In realtà

- è possibile disinteressarsi dell'efficienza durante la scrittura delle interrogazioni SQL
- l'utente scrive una interrogazione specificando quali operatori applicare, ma non in quale ordine
- il DBMS si incarica di effettuare le interrogazioni, trovando l'ordine di applicazione più efficiente per gli operatori

Forma Standard

◆ Metodo di scrittura delle interrogazioni

- Stabilire se è necessario utilizzare operatori insiemistici e dividere in sottointerrogazioni
- Per ogni sottointerrogazione, decidere da quali tabelle prelevare i dati
- Se le tabelle sono più di una,
 - (strategia a) metterle in prodotto cartesiano oppure
 - (strategia b) metterle in join con le condizioni opportune

Forma Standard

◆ Metodo di scrittura (continua)

- Scrivere le eventuali selezioni
 - (Strategia a) incluse le condizioni di Join
- Scrivere le eventuali proiezioni
- Scrivere le eventuali ridenominazioni finali
- Rimettere il tutto insieme applicando gli operatori insiemistici

Forma Standard

◆ Suggerimento n.1

- dare un nome a ciascun risultato parziale
- rimettere assieme gli operatori solo alla fine

◆ Suggerimento n.2

- tenere traccia dello schema dei risultati parziali generati da ciascun operatore
- elenco degli attributi (nomi e tipi)
- aiuta a scrivere le operazioni successive

Concetti Fondamentali

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

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

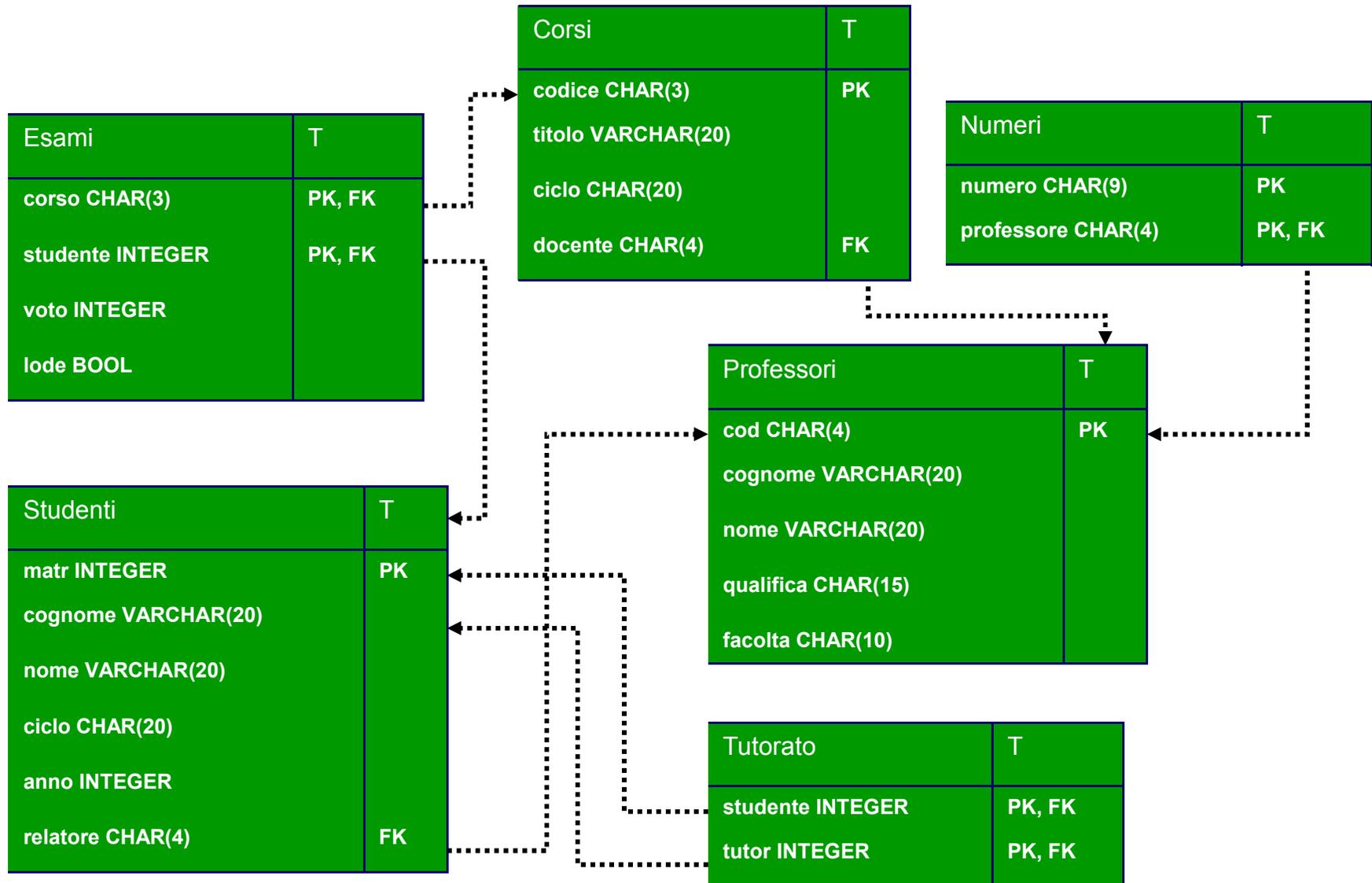
```
TABLE Studenti (
    matr integer PRIMARY KEY,
    cognome varchar(20) NOT NULL,
    nome varchar(20) NOT NULL,
    ciclo char(20),
    anno integer,
    relatore char(4)
    REFERENCES Professori(cod)
);
```

```
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));
```



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

<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>	<u>voto</u>	<u>lode</u>
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