



# Gestione e Analisi dei Dati

Lezione 2

Vincoli su attributo

Selezioni semplici su una tabella

# Vincoli sui Dati

- Regole del dominio di interesse
  - Unicità di codici di corso e matricole
    - Un codice deve corrispondere ad un solo corso
    - Una matricola deve corrispondere ad un solo studente
  - Voti degli studenti
    - da 18 a 30
    - lode solo se il voto è 30
- Correttezza dei riferimenti

# Una base di dati con errori

## Studenti

| Matricola | Cognome | Nome    | Data Nascita |
|-----------|---------|---------|--------------|
| 6554      | Pinco   | Pallino | 05/12/1978   |
| 8765      | Neri    | Paolo   | 03/11/1976   |
| 9283      | Verdi   | Luisa   | 12/11/1979   |
| 9283      | Rossi   | Maria   | 01/02/1978   |

Unicità della matricola

## Esami

| Studente | Voto | Lode | Corso |
|----------|------|------|-------|
| 3456     | 34   | 1    | 04    |
| 3456     | 24   | 0    | 02    |
| 9283     | 28   | 1    | 01    |
| 1111     | 26   | 0    | 01    |

Voti scorretti

Riferimento scorretto

# Vincoli di Integrità

- Regole imposte sui valori della base di dati
- Vincoli sulle singole tabelle
  - vincoli di chiave
  - vincoli di ennupla
- Vincoli tra tabelle diverse
  - vincoli di riferimento o di integrità referenziale

# Vincoli di Integrità

- Vincoli di chiave
  - chiave: identificatore per le ennuple
  - es: “matricola” è una chiave per “Studenti”
- Vincoli di ennupla
  - predicati sui valori delle ennuple
  - es:  $(\text{voto} \geq 18 \text{ and } \text{voto} \leq 30)$
- Vincoli di Riferimento
  - assenza di riferimenti inesistenti
  - es: esistono esami solo per gli studenti della base di dati

# Vincoli di Chiave

| Studenti | Matricola | Cognome | Nome    | Data Nascita |
|----------|-----------|---------|---------|--------------|
|          | 6554      | Pinco   | Pallino | 05/12/1978   |
|          | 8765      | Neri    | Paolo   | 03/11/1976   |
|          | 9283      | Verdi   | Luisa   | 12/11/1979   |
|          | 3456      | Rossi   | Maria   | 01/02/1978   |

```
CREATE TABLE Studenti(matricola integer,  
                        cognome char(20),  
                        nome char(20),  
                        dataNascita date,  
                        UNIQUE(matricola));
```

*oppure:*

```
TABLE Studenti(matricola integer PRIMARY KEY, ...
```

# Vincoli di Ennupla

| Esami | Studente | Voto | Lode | Corso |
|-------|----------|------|------|-------|
|       | 3456     | 30   | 1    | 04    |
|       | 3456     | 24   | 0    | 02    |
|       | 9283     | 28   | 0    | 01    |
|       | 6554     | 26   | 0    | 01    |

```
TABLE Esami (matricola integer,  
voto integer,  
corso char(3),  
lode bool,  
CHECK (voto>=18 and voto<=30),  
CHECK (not lode or voto=30));
```

# Vincoli di Riferimento

| Esami | Studente | Voto | Lode | Corso |
|-------|----------|------|------|-------|
|       | 3456     | 30   | 1    | 04    |
|       | 3456     | 24   | 0    | 02    |
|       | 9283     | 28   | 0    | 01    |
|       | 6554     | 26   | 0    | 01    |

```
CREATE TABLE Esami (matricola integer,  
                    corso char(3), voto integer, lode bool,  
                    CHECK (voto>=18 and voto<=30),  
                    CHECK (not lode or voto=30),  
                    FOREIGN KEY(matricola)  
                    REFERENCES Studenti(matricola),  
                    FOREIGN KEY(corso)  
                    REFERENCES Corsi(codice) );
```

“chiave esterna” =  
riferimento ad un  
attributo chiave di  
un'altra tabella

*oppure:*

```
CREATE TABLE Esami (matricola integer  
                    REFERENCES Studenti(matricola), ...
```

```
TABLE Studenti (matricola integer,  
                cognome char(20),  
                nome char(20),  
                dataNascita date,  
                UNIQUE(matricola));
```

```
TABLE Corsi (codice char(3),  
            titolo char(50),  
            docente char(20),  
            UNIQUE(codice));
```

```
TABLE Esami (matricola integer,  
            corso char(3),  
            voto integer,  
            lode bool,  
            CHECK (voto>=18 and voto<=30),  
            CHECK (not lode or voto=30),  
            FOREIGN KEY(matricola)  
                REFERENCES studenti(matricola),  
            FOREIGN KEY(corso)  
                REFERENCES corsi(codice),  
            UNIQUE (matricola, corso));
```

Schema  
con  
vincoli di  
integrità

# Caratteristiche del Modello

- Legami tra i dati basati sui valori
  - L'informazione è organizzata in più tabelle: la relazione tra tabelle differenti determina informazioni più complesse
- I valori devono essere semplici
  - valori “atomici” : numeri, caratteri, stringhe, booleani, date ecc.
  - non sono consentite “nidificazioni”

# Un Esempio di Informazione Nidificata

| <b><i>Dal Sudicio<br/>Via Buia, Pisa</i></b>           |                  |              |
|--|------------------|--------------|
| <b><i>Ricevuta Fiscale<br/>1235 del 12/10/2001</i></b> |                  |              |
| <b>3</b>   | <b>Coperti</b>   | <b>3,00</b>  |
| <b>2</b>   | <b>Antipasti</b> | <b>6,20</b>  |
| <b>3</b>   | <b>Primi</b>     | <b>12,00</b> |
| <b>2</b>   | <b>Bistecche</b> | <b>18,00</b> |
|  |                  |              |
|  |                  |              |
| <b><i>Totale</i></b>                                   |                  | <b>39,20</b> |

| <b><i>Dal Sudicio<br/>Via Buia, Pisa</i></b>           |                  |              |
|--|------------------|--------------|
| <b><i>Ricevuta Fiscale<br/>1240 del 13/10/2001</i></b> |                  |              |
| <b>2</b>   | <b>Coperti</b>   | <b>2,00</b>  |
| <b>2</b>   | <b>Antipasti</b> | <b>7,00</b>  |
| <b>2</b>   | <b>Primi</b>     | <b>8,00</b>  |
| <b>2</b>   | <b>Orate</b>     | <b>20,00</b> |
| <b>2</b>   | <b>Caffè</b>     | <b>2,00</b>  |
|  |                  |              |
| <b><i>Totale</i></b>                                   |                  | <b>39,00</b> |

Come organizzare l'informazione in modo da gestire i dettagli nidificati dentro ogni ricevuta?

# Una possibile rappresentazione

|   |   |
|---|---|
| <b>Dal Sudicio<br/>Via Buia, Pisa</b>           |   |
| <b>Ricevuta Fiscale<br/>1235 del 12/10/2001</b> |   |
| 3   | <b>Dal Sudicio<br/>Via Buia, Pisa</b>           |
| 2   | <b>Dal Sudicio<br/>Via Buia, Pisa</b>           |
| 3   | <b>Ricevuta Fiscale<br/>1240 del 13/10/2001</b> |
| 2   | <b>Ricevuta Fiscale<br/>1240 del 13/10/2001</b> |
| 2   | Coperti 2,00                                    |
| 2   | Antipasti 7,00                                  |
| 2   | Primi 8,00                                      |
| 2   | Orate 20,00                                     |
| 2   | Caffè 2,00                                      |
|   |   |
| <b>Totale 39,00</b>                             |   |

Una tabella gestisce la lista di tutte le ricevute

| Numero | Data       | Totale |
|--------|------------|--------|
| 1235   | 12/10/2001 | 39,20  |
| 1240   | 13/10/2001 | 39,00  |

I dettagli di ogni ricevuta sono contenuti in una tabella, collegata tramite chiave esterna

| Numero | Qta | Descriz   | Prezzo |
|--------|-----|-----------|--------|
| 1235   | 3   | Coperti   | 3,00   |
| 1235   | 2   | Antipasti | 6,20   |
| 1235   | 3   | Primi     | 12,00  |
| 1235   | 2   | Bistecche | 18,00  |
| 1240   | 2   | Coperti   | 2,00   |
| 1240   | 2   | Antipasti | 7,00   |
| ...    | ... | ...       | ...    |

# Basi di dati

| <b>Studenti</b> | <b>Matricola</b> | <b>Cognome</b> | <b>Nome</b> | <b>Data Nascita</b> |
|-----------------|------------------|----------------|-------------|---------------------|
|                 | 6554             | Pinco          | Pallino     | 05/12/1978          |
|                 | 8765             | Neri           | Paolo       | 03/11/1976          |
|                 | 9283             | Verdi          | Luisa       | 12/11/1979          |
|                 | 3456             | Rossi          | Maria       | 01/02/1978          |

| <b>Corsi</b> | <b>Codice</b> | <b>Titolo</b> | <b>Docente</b> |
|--------------|---------------|---------------|----------------|
|              | a01           | Analisi       | Sempronio      |
|              | c02           | Chimica       | Bruni          |
|              | c04           | Chimica       | Verdi          |

| <b>Esami</b> | <b>Studente</b> | <b>Voto</b> | <b>Lode</b> | <b>Corso</b> |
|--------------|-----------------|-------------|-------------|--------------|
|              | 3456            | 30          | 1           | c04          |
|              | 3456            | 24          | 0           | c02          |
|              | 9283            | 28          | 0           | a01          |
|              | 6554            | 26          | 0           | a01          |

Data questa organizzazione, sono necessari altri strumenti per correlare tra loro le tabelle

# 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

# Esempio: base di dati per corso 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

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

```
CREATE 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)  
);
```

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

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

```
CREATE 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));
```

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

## Professori

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

## 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       |
| 777  | Bruno   | Pasquale | laurea sp. | 1    | FT       |
| 888  | Pinco   | Pietro   | laurea sp. | 1    | CV       |

## Corsi

| cod  | titolo                | ciclo      | docente |
|------|-----------------------|------------|---------|
| PR1  | Programmazione I      | laurea tr. | FT      |
| ASD  | Algoritmi e Str. Dati | laurea tr. | CV      |
| INFT | Informatica Teorica   | laurea sp. | ADP     |

## Numeri

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

## Tutorato

| studente | tutor |
|----------|-------|
| 111      | 777   |
| 222      | 777   |
| 333      | 888   |
| 444      | 888   |

## Esami

| studente | corso | voto | lode  |
|----------|-------|------|-------|
| 111      | PRI   | 27   | false |
| 222      | ASD   | 30   | true  |
| 111      | INFT  | 24   | false |
| 777      | PRI   | 21   | false |
| 777      | ASD   | 28   | false |
| 888      | ASD   | 28   | false |
| 888      | PRI   | 30   | false |
| 888      | INFT  | 30   | true  |

# Algebra Relazionale

- **Operatori principali**
  - selezione
  - proiezione
  - prodotto cartesiano e join
  - unione, intersezione e differenza

# 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 dei soli studenti della laurea triennale”

# Studenti laurea triennale

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       |
| 777  | Bruno   | Pasquale | laurea sp.   | 1    | FT       |
| 888  | Pinco   | Pietro   | laurea sp.   | 1    | CV       |

**condizione: Studenti.ciclo='laurea tr.'**

Studenti  
Triennale

| 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       |

Il risultato della selezione è una tabella contenente solo le ennuple che soddisfano la condizione

# 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:

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

# Proiezione

- Funzione
  - estrarre alcune delle colonne di una tabella
- Esempio
  - “Estrarre l’elenco dei nomi e i cognomi degli studenti”

```
CREATE 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),  
);
```

# Proiezione: cognome e 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       |
| 777  | Bruno   | Pasquale | laurea sp. | 1    | FT       |
| 888  | Pinco   | Pietro   | laurea sp. | 1    | CV       |

ElencoNomi

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

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

# Proiezione: sintassi

- Operatore unario
  - lista di attributi della tabella
- Sintassi  $\pi_{\text{lista attributi}}(R)$
- Esempio d'uso

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

# Linguaggio SQL

- **Interrogazioni SQL**
  - una o più sotto-interrogazioni
  - correlate da operatori insiemistici
- **Sotto-interrogazioni: varie “clausole”**
- **Nucleo della SELECT**
  - SELECT: proiezioni, distinct
  - FROM: prodotti cartesiani o join, alias
  - [WHERE]: selezioni
- **Clausole aggiuntive**
  - [ORDER BY]: ordinamenti

# SQL: Selezione

- Studenti della laurea triennale di anni successivi al primo
  - $\sigma_{\text{ciclo}='laurea\ tr.' \text{ AND } \text{anno}>1}$  (Studenti)
  - **SELECT** \*  
**FROM** Studenti  
**WHERE** ciclo='laurea tr.' AND anno>1;

# SQL: Proiezione e DISTINCT

- Cognomi e nomi degli studenti, senza duplicati
  - ElencoNomi =  $\text{DISTINCT}(\pi_{\text{cognome, nome}}(\text{Studenti}))$
  - **SELECT** **DISTINCT** **cognome, nome**  
**FROM** **Studenti**;