

# SQL

## **Laboratorio di** ***Basi di Dati*** (CdSin Informatica) ***Progettazione di Basi di Dati*** (CdSin Informatica e TPS)

**a.a. 2010/2011**

<http://www.di.uniba.it/~lisi/courses/basi-dati/bd2010-11.htm>

dott.ssa Francesca A. Lisi  
lisi@di.uniba.it

Orario di ricevimento: mercoledì ore 10-12

# Sommario (II parte)

- Manipolazione dei dati in SQL
  - inserimento di tuple
  - cancellazione di tuple
  - modifica di tuple
- Script SQL

## Riferimenti

- cap. 2 di Pratt
- cap. 5 di “MySQL Tutorial”
- cap. 8, in particolare 8.6, di Elmasri & Navathe
- cap. 4, in particolare 4.4, di Atzeni et al.

**Maternità**

<b>Madre</b>	<b>Figlio</b>
Luisa	Maria
Luisa	Luigi
Anna	Olga
Anna	Filippo
Maria	Andrea
Maria	Aldo

**Paternità**

<b>Padre</b>	<b>Figlio</b>
Sergio	Franco
Luigi	Olga
Luigi	Filippo
Franco	Andrea
Franco	Aldo

**Persone**

<b>Nome</b>	<b>Età</b>	<b>Reddito</b>
Andrea	27	21
Aldo	25	15
Maria	55	42
Anna	50	35
Filippo	26	30
Luigi	50	40
Franco	60	20
Olga	30	41
Sergio	85	35
Luisa	75	87

# Inserimento

```
INSERT INTO NomeTabella  
[ (ListaAttributi) ] VALUES (ListaValori)
```

```
INSERT INTO Persone  
      VALUES ('Mario', 25, 52)
```

```
INSERT INTO Persone(Nome, Eta, Reddito)  
      VALUES ('Pino', 25, 52)
```

```
INSERT INTO Persone(Nome, Reddito)  
      VALUES ('Lino', 55)
```

## Inserimento (II)

- l'ordinamento degli attributi (se presente) e dei valori è significativo
- le due liste debbono avere lo stesso numero di elementi
- se la lista di attributi è omessa, si fa riferimento a tutti gli attributi della relazione, secondo l'ordine con cui sono stati definiti
- se la lista di attributi non contiene tutti gli attributi della relazione, per gli altri viene inserito un valore nullo (che deve essere permesso) o un valore di default

# Cancellazione

**DELETE FROM** *NomeTabella*  
[**WHERE** *Condizione*]

- elimina le tuple che soddisfano la condizione
- può causare (se i vincoli di integrità referenziale sono definiti con politiche di reazione **CASCADE**) la cancellazione di tuple da altre relazioni
- ricordare: se la clausola **WHERE** viene omessa, si intende **WHERE TRUE**

## Cancellazione (II)

```
DELETE FROM Persone WHERE Eta < 35
```

```
DELETE FROM Paternita
```

Cancellazione di tabella (anche non vuota)

```
DROP TABLE Paternita CASCADE
```

Cancellazione di tabella (solo se vuota - *default*)

```
DROP TABLE Paternita RESTRICT
```

# Aggiornamento

**UPDATE** *NomeTabella*

**SET** *NomeAttributo* =

*< Espressione | Select-query | NULL | DEFAULT >*

[ **WHERE** *Condizione* ]

**UPDATE** Persone

**SET** Reddito = 45 **WHERE** Nome = 'Piero'

**UPDATE** Persone

**SET** Reddito = Reddito\*1.1

**WHERE** Eta < 30



# Aggiornamento (II)

```
CREATE TABLE Impiegato (  
    Matricola CHAR(6) PRIMARY KEY,  
    Nome CHAR(20) NOT NULL,  
    Cognome CHAR(20) NOT NULL,  
    Dip CHAR(15)  
        REFERENCES Dipartimento(Nome)  
        ON DELETE SET NULL  
        ON UPDATE CASCADE,  
    Ufficio NUMERIC(3),  
    Stipendio NUMERIC(9) DEFAULT 0,  
    UNIQUE (Cognome, Nome)  
)
```

# Il database Prodotti Premiere

- Popolare la tabella Rappresentanti con le seguenti tuple

03		Jones		Mary		123 Main		Grant		MI		49219		2150.00		.05
18		Martin		Elyse												

```
INSERT INTO Rappresentanti VALUES
  ('03', 'Jones', 'Mary', '123 Main', 'Grant',
   'MI', '49219', 2150.00, .05);
```

```
INSERT INTO Rappresentanti (CodRappr, Cognome,
  Nome) VALUES ('18', 'Martin', 'Elyse');
```

# Il database Prodotti Premiere (II)

- Modificare il valore dell'attributo `Cognome` della tupla `'18'` della tabella `Rappresentanti` nel modo seguente

18		Marlin		Elyse											
----	--	--------	--	-------	--	--	--	--	--	--	--	--	--	--	--

```
UPDATE Rappresentanti SET Cognome = 'Marlin'  
WHERE CodRappr = '18';
```

- Cancellare la tupla `'18'` dalla tabella `Rappresentanti`

```
DELETE FROM Rappresentanti  
WHERE CodRappr = '18';
```

# Script SQL

- I comandi SQL necessari alla creazione ed al popolamento di un database possono essere raccolti in uno o più file di testo (detti *script*) e successivamente fatti eseguire da un interprete SQL

## Esercizio

- Scrivere con Notepad uno script per la creazione ed il popolamento delle tabelle del database Prodotti Premiere
- Salvare lo script in formato testo ASCII con il nome  
`premiere-db.sql`

# Esercitazione con MySQL

- Creare ed utilizzare il database Prodotti Premiere

```
mysql> CREATE DATABASE premiere;
```

```
mysql> USE premiere;
```

- Creare e popolare le tabelle del database con uno script

```
mysql> SOURCE premiere-db.sql;
```

- Verificare l'avvenuta creazione delle tabelle

```
mysql> SHOW TABLES;
```

```
mysql> DESCRIBE Rappresentanti;
```

...

- Verificare l'avvenuto popolamento delle tabelle

```
mysql> SELECT * FROM Rappresentanti;
```

# Esercitazione con MySQL (II)

- Definire politiche di reazione alla violazione del vincolo di integrità referenziale per ogni chiave esterna, p.es.

```
CREATE TABLE Ordini (  
  NroOrdine CHAR(5) PRIMARY KEY,  
  Data DATE,  
  CodCliente CHAR(3),  
  INDEX Cli_ind (CodCliente),  
  FOREIGN KEY (CodCliente) REFERENCES  
    Clienti(CodCliente) ON DELETE SET NULL  
) ENGINE=INNODB;
```

- Provare comandi di cancellazione e modifica di tuple nelle tabelle referenziate per verificare in che misura MySQL 5.1 supporta l'integrità referenziale.