

# SQL

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

**a.a. 2015/2016**

<http://www.di.uniba.it/~lisi/courses/basi-dati/bd2015-16.htm>

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

Orario di ricevimento: giovedì 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

- Usando un editore di testo, creare uno script con tutti i comandi di inserimento tuple nelle tabelle della base dati `premiere`
- Salvare lo script in formato testo ASCII con il nome  
`premiere_db_instance.sql`

# Esercitazione con MySQL

- Dalla shell di MySQL lanciare lo script di creazione dello schema della base dati `premiere` con la configurazione MyISAM di MySQL (messo a disposizione dal docente)

```
mysql> SOURCE premiere_db_schema_MyISAM.sql;
```

- Verificare l'avvenuta creazione della base dati `premiere`
- Dalla shell di MySQL lanciare lo script di popolamento

```
mysql> SOURCE premiere_db_instance.sql;
```

- Verificare l'avvenuto popolamento delle tabelle

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

...

- Provare i comandi di cancellazione e modifica di tuple sulla base dati `premiere` così popolata

# Esercitazione con MySQL (II)

- Creare uno script di creazione dello schema della base dati `premiere` con la configurazione InnoDB di MySQL , definendo politiche di reazione alla violazione del vincolo di integrità referenziale per ogni chiave esterna
- Salvare lo script con il nome  
`premiere_db_schema_InnoDB.sql`
- Lanciarlo dalla shell di MySQL
- Lanciare lo script `premiere_db_instance.sql`
- Provare i comandi di cancellazione e modifica di tuple sulla versione InnoDB della base dati `premiere`