

# 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

# Cos'è SQL?

- Linguaggio di riferimento per le basi di dati relazionali
- acronimo di *Structured Query Language*
- comprende sia un DDL che un DML
- un po' di storia
  - prima proposta **SEQUEL** (1974);
  - prime implementazioni in SQL/DS e Oracle (1981)
  - dal 1983 ca. "standard de facto" dei linguaggi per basi di dati relazionali
  - ANSI-ISO standard (1986, 1989, 1992, 1999)
    - recepito solo in parte (!!)

# Pre-requisiti ed obiettivi

- Linguaggi per basi di dati
  - Modello relazionale dei dati
  - Esperienze di programmazione
- 
- Operatività in tempi ridotti
  - Approfondimenti *in itinere*

# Testi adottati

- “Guida ad SQL”  
di P.J. Pratt  
edito da Apogeo, 2001
- “MySQL Tutorial”  
di L. Welling, L. Thomson  
edito da Pearson Education Italia, 2004
- Dispense a cura del docente
  - <http://www.di.uniba.it/~lisi/courses/basi-dati/bd2010-11.htm>

# Testi consigliati

- “Sistemi di Basi di Dati - Fondamenti” (cap. 8)  
di R.A. Elmasri, S.B. Navathe  
edito da Pearson Education Italia, 2004
- “Basi di dati - Modelli e linguaggi di  
interrogazione (III ed.)” (capp. 4-5)  
di P. Atzeni, S. Ceri, S. Paraboschi, R. Torlone  
edito da McGraw-Hill, 2009

# Sommario (I parte)

- Definizione dei dati in SQL
  - Definizione di schema
  - Definizione di domini
  - Definizione di tabelle
  - Definizione di vincoli
  - Modifica dello schema
- Primi passi con MySQL

## Riferimenti

- capp. 1-2 di Pratt
- capp. 1-2, 4-5 di “MySQL Tutorial”
- cap. 8, in particolare 8.1-8.3, di Elmasri & Navathe
- cap. 4, in particolare 4.1-4.2, di Atzeni et al.

# Definizione di schemi

- Istruzione **CREATE SCHEMA**:
  - definisce uno schema di base di dati come collezione di oggetti
  - crea un'istanza vuota dello schema

```
CREATE SCHEMA Premiere
```

# Domini elementari

- **CHARACTER**: singoli caratteri o stringhe, anche di lunghezza variabile
- **BIT**: singoli booleani o stringhe
- **NUMERIC, DECIMAL, INTEGER, SMALLINT**: tipi numerici esatti
- **FLOAT, REAL, DOUBLE PRECISION**: tipi numerici approssimati
- **DATE, TIME, TIMESTAMP**: istanti temporali
- **INTERVAL**: intervalli di tempo



# Definizione di domini

- Istruzione **CREATE DOMAIN**:
  - definisce un dominio (semplice), utilizzabile in definizioni di relazioni, anche con vincoli e valori di default

```
CREATE DOMAIN Voto  
AS SMALLINT DEFAULT NULL  
CHECK (VALUE >=18 AND VALUE <=30)
```

# Definizione di tabelle

- Istruzione **CREATE TABLE**:
  - definisce uno schema di relazione e ne crea un'istanza vuota
  - specifica attributi, domini e vincoli

Dipartimento(Nome, Indirizzo, Sede)

```
CREATE TABLE Dipartimento (  
    Nome CHAR(20) PRIMARY KEY,  
    Indirizzo CHAR(50),  
    Sede CHAR(20)  
)
```

# Vincoli intrarelazionali

- **NOT NULL**
- **UNIQUE** definisce le chiavi di una relazione
- **PRIMARY KEY** definisce la chiave primaria di una relazione (implica **NOT NULL**)
- **CHECK** esprime vincoli di integrità (ne parleremo più avanti)

Impiegato (Matricola, Nome, Cognome, Dip, Stipendio)

# Definizione di chiavi interne

```
CREATE TABLE Impiegato(  
    Matricola CHAR(6) PRIMARY KEY,  
    Nome CHAR(20) NOT NULL,  
    Cognome CHAR(20) NOT NULL,  
    Dip CHAR(15),  
    Stipendio NUMERIC(9) DEFAULT 0,  
    FOREIGN KEY (Dip)  
        REFERENCES Dipartimento(Nome),  
    UNIQUE (Cognome, Nome)  
)
```

# Definizione di chiavi interne (II)

Matricola **CHAR(6) PRIMARY KEY**

Matricola **CHAR(6) ,**

**... ,**

**PRIMARY KEY (Matricola)**

- Sono equivalenti!

# Definizione di chiavi interne (III)

Nome      **CHAR(20) NOT NULL,**  
Cognome **CHAR(20) NOT NULL,**  
**UNIQUE (Cognome, Nome),**

Nome      **CHAR(20) NOT NULL UNIQUE,**  
Cognome **CHAR(20) NOT NULL UNIQUE,**

- Non sono equivalenti!

# Vincoli interrelazionali

- **REFERENCES** e **FOREIGN KEY** permettono di definire *vincoli di integrità referenziale*
- di nuovo due sintassi
  - per singoli attributi
  - su più attributi
- E' possibile definire politiche di reazione alla violazione dell'integrità referenziale
- **CHECK**, vedremo più avanti

# Infrazioni

<u>Codice</u>	Data	Vigile	Prov	Numero
34321	1/2/95	3987	MI	39548K
53524	4/3/95	3295	TO	E39548
64521	5/4/96	3295	PR	839548
73321	5/2/98	9345	PR	839548

## Vigili

<u>Matricola</u>	Cognome	Nome
3987	Rossi	Luca
3295	Neri	Piero
9345	Neri	Mario
7543	Mori	Gino



# Infrazioni

<u>Codice</u>	Data	Vigile	Prov	Numero
34321	1/2/95	3987	MI	39548K
53524	4/3/95	3295	TO	E39548
64521	5/4/96	3295	PR	839548
73321	5/2/98	9345	PR	839548

## Auto

<u>Prov</u>	<u>Numero</u>	Cognome	Nome
MI	39548K	Rossi	Mario
TO	E39548	Rossi	Mario
PR	839548	Neri	Luca

# Definizione di chiavi esterne

```
CREATE TABLE Infrazioni (  
    Codice CHAR(6) PRIMARY KEY,  
    Data DATE NOT NULL,  
    Vigile INTEGER NOT NULL  
        REFERENCES Vigili(Matricola),  
    Provincia CHAR(2),  
    Numero CHAR(6) ,  
    FOREIGN KEY (Provincia, Numero)  
        REFERENCES Auto(Provincia,  
    Numero)
```

# Modifiche degli schemi

- Istruzione **ALTER**:
  - consente di modificare domini e schemi di tabelle

```
ALTER TABLE Dipartimento  
ADD COLUMN NroUff NUMERIC(4)
```

# Modifiche degli schemi

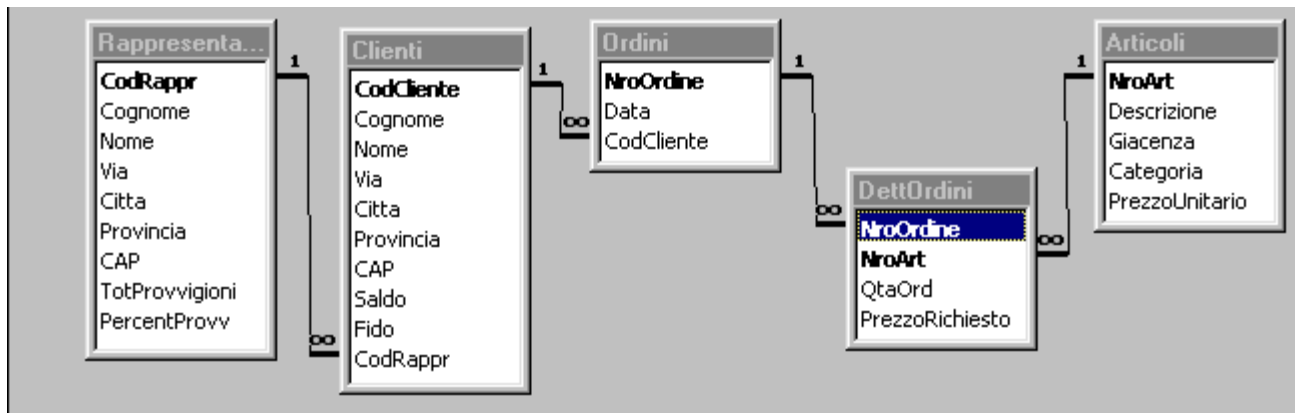
- Istruzione **DROP**:
  - consente di rimuovere componenti dello schema
  - opzione **RESTRICT** (*default!*) specifica che il comando non deve essere eseguito su oggetti non vuoti
  - opzione **CASCADE** specifica che il comando deve essere applicato su oggetti anche non vuoti

**DROP TABLE** Dipartimento

**DROP TABLE** Dipartimento **CASCADE**

# Il database Prodotti Premiere

- Cfr. “Guida ad SQL” cap. 2



- Creare lo schema della base di dati
- Definire le tabelle, dimensionando opportunamente gli attributi ed indicando i vincoli intra-/inter-relazionali

# Il database Prodotti Premiere (II)

Rappresentanti(CodRappr, Cognome, Nome, Via, Citta, Provincia, CAP, TotProvvigioni, PercentProv) )

```
CREATE TABLE Rappresentanti (  
    CodRappr CHAR(2) PRIMARY KEY,  
    Cognome CHAR(10) NOT NULL,  
    Nome CHAR(8) NOT NULL,  
    Via CHAR(15),  
    Citta CHAR(15),  
    Provincia CHAR(2),  
    CAP CHAR(5),  
    TotProvvigioni DECIMAL(7,2),  
    PercentProv DECIMAL(3,2) );
```

# Il database Prodotti Premiere (III)

Clienti (CodCliente, Cognome, Nome, Via, Citta, Provincia, CAP, Saldo, Fido, CodRappr)

```
CREATE TABLE Clienti (
    CodCliente CHAR(3) PRIMARY KEY,
    Cognome CHAR(10) NOT NULL,
    Nome CHAR(8) NOT NULL,
    Via CHAR(15),
    Citta CHAR(15),
    Provincia CHAR(2),
    CAP CHAR(5),
    Saldo DECIMAL(7,2) DEFAULT 0,
    Fido DECIMAL(6,2) DEFAULT 0,
    CodRappr CHAR(2)
    REFERENCES Rappresentanti(CodRappr) );
```

# Il database Prodotti Premiere (IV)

Ordini (NroOrdine, Data, CodCliente)

```
CREATE TABLE Ordini (  
    NroOrdine CHAR(5) PRIMARY KEY,  
    Data DATE,  
    CodCliente CHAR(3)  
        REFERENCES Clienti(CodCliente)  
);
```



# Il database Prodotti Premiere (V)

Articoli(NroArt, Descrizione, Giacenza, Categoria, PrezzoUnitario)

```
CREATE TABLE Articoli (  
    NroArt CHAR(4) PRIMARY KEY,  
    Descrizione CHAR(20),  
    Giacenza DECIMAL(4,0),  
    Categoria CHAR(2),  
    PrezzoUnitario DECIMAL(6,2)  
);
```

# Il database Prodotti Premiere (VI)

DettOrdini (NroOrdine, NroArt, QtaOrd, PrezzoRichiesto)

```
CREATE TABLE DettOrdini (  
    NroOrdine CHAR(6)  
        REFERENCES Ordini(NroOrdine),  
    NroArt CHAR(4)  
        REFERENCES Articoli(NroArt),  
    PRIMARY KEY (NroOrdine, NroArt),  
    QtaOrd DECIMAL(3,0),  
    PrezzoRichiesto DECIMAL(6,2)  
);
```

## II DBMS MySQL

<http://www.mysql.com/>

Versione stabile corrente: MySQL 5.1

- DBMS con architettura client-server
- Installare e lanciare il server (vedi manuale utente)
- Connettersi al server

```
DOS-shell> mysql -u user_name -p
```

..in modalità batch

```
..
```

```
..
```

- Disconnettersi dal server

```
mysql> QUIT
```

## II DBMS MySQL (II)

- Creare un database

```
mysql> CREATE DATABASE db_name;
```

- Selezionare un database esistente

```
mysql> USE db_name;
```

- Interagire con il database selezionato

```
mysql> SHOW TABLES;
```

```
mysql> CREATE TABLE t1 (...);
```

```
mysql> DESCRIBE t1;
```

...

- Impostare la modalità batch (uso di script SQL)

```
mysql> SOURCE file_name;
```

## II DBMS MySQL (III)

- Integrità referenziale non è supportata per default in MySQL 5.1 ma è ottenibile
  - usando tabelle di tipo InnoDB;
  - definendo indici sulle chiavi esterne.
- Per es., la tabella `Ordini` in `premiere` diventa:

```
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;
```

# Esercitazione con MySQL

- Creare ed utilizzare il database Prodotti Premiere

```
mysql> CREATE DATABASE Premiere;
```

```
mysql> USE Premiere;
```

- Creare le tabelle del database, per es.

```
mysql> CREATE TABLE Rappresentanti (..);
```

- Verificare l'avvenuta creazione delle tabelle

```
mysql> SHOW TABLES;
```

- Verificare la correttezza dello schema delle tabelle, per es.

```
mysql> DESCRIBE Rappresentanti;
```

- Provare i comandi di modifica dello schema
  - N.B. Aggiunta di chiavi esterne possibile solo su attributi già indicizzati