SQL

Laboratorio di Basi di Dati
a.a. 2002/2003

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 ***S**tructured **Q**uery **L**anguage*
- 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 consigliati

- “Basi di dati - Modelli e linguaggi di interrogazione”

  di P. Atzeni, S. Ceri, S. Paraboschi, R. Torlone

  edito da McGraw-Hill, 2002
- “Guida ad SQL”

  di P.J. Pratt

  edito da Apogeo, 2001
- Dispense a cura del docente
  - `www.di.uniba.it/~lisi/courses/basi-dati/`
  - biblioteca (I piano Dipartimento di Informatica)

Sommario (I parte)

- Definizione dei dati in SQL
  - I domini elementari
  - Definizione di domini
  - Definizione di tabelle
  - Vincoli intrarelazionali ed interrelazionali

Riferimenti

- cap. 4, in particolare 4.1, di Atzeni et al.
- capp. 1-2 di Pratt

Domini elementari

- Carattere: singoli caratteri o stringhe, anche di lunghezza variabile
- Bit: singoli booleani o stringhe
- Numerici, esatti e approssimati
- Data, ora, intervalli di tempo
- Introdotti in SQL:1999:
  - Boolean
  - BLOB, CLOB (binary/character large object): per grandi immagini e testi

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),
)
```

Modifiche degli schemi

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

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

- Istruzione **DROP**:
  - consente di rimuovere componenti dello schema

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

Codice	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

Matricola	Cognome	Nome
3987	Rossi	Luca
3295	Neri	Piero
9345	Neri	Mario
7543	Mori	Gino

Infrazioni

Codice	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

Prov	Numero	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)
 )
```

Esercizi

- Dare le definizioni SQL delle relazioni

Rappresentanti(CodRappr, Cognome, Nome, Via, Citta, Provincia, TotProvvigioni,PercentProvv)

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

Ordini (NroOrdine, Data, CodCliente)

DettOrdini (NroOrdine, NroArt, Qta, Prezzo)

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

dimensionando opportunamente gli attributi ed indicando i vincoli sia intrarelazionali che interrelazionali

- Esercizi 4.1- 4.6 da Atzeni et al., pagg. 159-160