

SQL: concetti di base

Fallucchi Francesca

fallucchi@info.uniroma2.it

<http://art.uniroma2.it/fallucchi/>

a.a. 2010/2011

SQL: introduzione

- Non è un linguaggio di programmazione
- Segue un paradigma dichiarativo: si specifica la descrizione dell'obiettivo
- l'interrogazione in SQL viene tradotta dai DBMS in linguaggio procedurale, quindi viene ottimizzata
- sulla traduzione si fanno ottimizzazioni algebriche ...
- ... e non (queste ultime dipendono dalle strutture sottostanti al DBMS in questione)

SQL: introduzione

- Permette di definire lo schema di un database, popolarne le tabelle, aggiornarne i dati, eseguire query, definire vincoli ed autorizzazioni.
- le implementazioni in genere non rispettano pienamente gli standard

SQL: introduzione

- Data Definition Language (DDL):
 - definizione dello schema del DB
- Data Manipulation Language (DML):
 - utilizzo dei dati del DB

Data Definition Language (DDL)

- E' il primo dei linguaggi di SQL ed è dedicato alla definizione della parte intensionale del database:
 - creazione del database
 - tabelle, campi e domini
 - indici
 - eventuali modifiche allo schema
 - vincoli, autorizzazioni

Creazione di un Data Base

- Sintassi:

```
create database <database>
[create-specs]
```
- se non vi sono "create-specs", il DBMS alloca un insieme di risorse predefinite per il nuovo database (e.g. character set, page size, etc..).
- Esempio:

```
CREATE DATABASE Azienda
```

Modifica di un Data Base

- Sintassi:

```
alter database <database>  
    <alter-specs>
```
- Modifica le impostazioni relative alle risorse fisiche impegnate dal database indicato
- Esempio:

```
ALTER DATABASE Azienda  
ADD FILE AziendaSPA.gdb
```

Cancellazione di un Data Base

- Sintassi:

```
drop database <database>
```
- Cancella tutte le tabelle e libera le risorse fisiche impegnate dal database indicato
- Esempio:

```
DROP DATABASE Azienda
```

Creare Schema

- Per passare alla definizione dello schema delle singole tabelle, SQL richiede l'ulteriore comando "CREATE SCHEMA"
- Sintassi:

```
Create schema[authorization owner]  
    [schema elements]
```
- "AUTHORIZATION" specifica il proprietario dello schema che si va definendo
- "schema elements" comprende la definizione dello schema delle singole tabelle (CREATE TABLE), i diritti di accesso (GRANT) e le viste (CREATE VIEW).

Creare una tabella

- una tabella è una collezione di attributi e può avere vincoli
- Ciascun attributo può, a sua volta, avere dei vincoli
- Istruzione CREATE TABLE :
 - definisce uno schema di relazione e ne crea un'istanza vuota
 - specifica attributi, domini e vincoli

Creare una tabella

- Sintassi:

```
create table NomeTabella (nomeAttr  
    dominio [default-value] [vincoli]  
    {, nomeAttr dominio [default-value]  
    [vincoli]} Atri vincoli)
```
- Inizialmente la tabella è vuota e il proprietario ha tutti i privilegi

Esempio creazione di una tabella

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

Modifica di una tabella

```
alter table nome-tab <operation>
[,<operation> ...]; <operation> =
{ADD [COLUMN] nome-col <col_def> |
ADD <constraint> | ALTER [COLUMN]
column_name <alt_col> | DROP col |
DROP CONSTRAINT constraint}
<alt_col> = {TO new_col_name | TYPE
new_col_datatype | POSITION
new_col_position}
```

Modifica di una tabella

- Esempio
alter table Dipartimento ADD NULL int

Cancellare una tabella

- Cancellazione di una tabella:
drop table nome-tab
- Esempio
drop table Dipartimento

Domini

- Un dominio è un insieme di valori ammissibili per un attributo (concetto simile a quello di “tipo”)
- Due possibilità in SQL
 - domini elementari (predefiniti)
 - domini definiti dall'utente (semplici, ma riutilizzabili)

Domini elementari

- SQL offre 6 famiglie di domini predefiniti:
 - Carattere
 - Bit
 - Tipi numerici esatti
 - Tipi numerici approssimati
 - Data e ora
 - Intervalli temporali

Domini definiti dall'utente

- si possono creare/manipolare domini nuovi a partire da quelli di base
- Si costruisce come un sottoinsieme di un dominio predefinito che soddisfi specifiche condizioni e vincoli:

```
create domain nome-dom as tipo
[default value][NOT NULL]
[check vincoli] default check
```

Domini definiti dall'utente

- Esempio

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

Modifica dominio definito da utente

- modifica di un dominio definito da un utente:

```
alter domain nome-dom set
[default value][NOT NULL]
[check vincoli] default check
```

- Esempio

```
alter domain annoCorso set type int
```

Cancellare dominio definito da utente

- cancellazione di un dominio definito da un utente:

```
drop domain nome-dom
```

- Esempio

```
drop domain annoCorso
```

Vincoli Intrarelazionali

- su tabelle e domini: specificano proprietà che ogni istanza del DB deve verificare

- vincoli predefiniti:

- not null: non può assumere valore nullo
- unique: attributo (insieme di attributi unique(attr{,attr})) è superchiave
- primary key: attributi come chiave primaria (implica il not null)

- vincoli non predefiniti:

- check

Vincolo Interrelazionale

- **foreign key**: chiave esterna

- impone che su ogni riga della tabella corrente (interna) il valore degli attributi specificati, se diversi da null, sia presente nelle tuple della tabella esterna tra i valori dei corrispondenti attributi.
- La sintassi impone che l'insieme di attributi della tabella esterna sia unique, cioè superchiave (vincolo di integrità referenziale)
- **references**: foreign key su singolo attributo

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

Prov	Numero	Cognome	Nome
MI	39548K	Rossi	Mario
TO	E39548	Rossi	Mario
PR	839548	Neri	Luca

Atzeni-Ceri-Paraboschi-Torfone, Basi di dati

CREATE TABLE, esempio

```
CREATE TABLE Infrazioni(  
  Codice CHAR(6) NOT NULL 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)  
)
```

Atzeni-Ceri-Paraboschi-Torfone, Basi di dati

Gestione Vincoli

- **Vincoli intrarelazionali:** il DBMS verifica che per ogni istanza (inserimento/modifica) i vincoli non siano violati. In caso di violazione, l'aggiunta/modifica viene rifiutata con un messaggio di errore
- **Vincoli interrelazionali:** non tutte le violazioni vengono trattate nello stesso modo ⇒ politiche di reazione alle violazioni

Gestione Vincolo Integrità Referenziale

- violazione su **tabella interna** (inserimento/modifica riga)
⇒ DBMS rifiuta l'operazione
- violazione su **tabella esterna** (cancellazione/modifica riga a cui fa riferimento tabella interna)
⇒ l'esterna è la tabella principale
⇒ si adottano politiche specifiche

Gestione Vincolo Integrità Referenziale

- **Modifica**
 - **Cascade:** valore di tabella esterna riportato su tutte tabelle interne
 - **set null:** inserisce null come valore degli attributi corrispondenti nelle tabelle interne
 - **set default:** inserisce il valore di default
 - **no action:** non si permette la modifica alla tabella esterna

Gestione Vincolo Integrità Referenziale

- **Cancellazione**
 - **Cascade:** cancellazione delle righe corrispondenti anche dalle tabelle interne
 - **set null:** inserisce null come valore degli attributi nelle tabelle interne corrispondenti ai valori delle righe cancellate nella tabella esterna
 - **set default:** inserisce il valore di default
 - **no action:** non si permette la cancellazione nella tabelle esterna

Specifica di politica di reazione

- Sintassi

```
on <delete | update>  
<cascade | set null |  
set default | no action>
```

Esempio specifica di politica di reazione

```
create table Impiegati (  
  Matr character(6) primary key,  
  Cogn character(20) not null,  
  Dip character(15),  
  Stip numeric(6) default 0,  
  unique(Cogn,Nome)  
  foreign key(Dip) references  
  Dipartimento(NomeDip)  
  on delete set null  
  on update cascade )
```

Definizione degli indici

- è rilevante dal punto di vista delle prestazioni
- ma è a livello fisico e non logico
- in passato era importante perché in alcuni sistemi era l'unico mezzo per definire chiavi

```
CREATE INDEX
```