Laboratorio di Basi di dati 1
Soluzioni della II esercitazione - 18 marzo 2003

La sintassi deri comandi T-SQL non è rigorosa, poiché permette di ottenere lo stesso risultato con diversi combinazioni di argomenti. Di seguito saranno indicate solo le possibilità principali.

1. • Per i gruppi che hanno creato, durante la precedente esercitazione, il database per la gestione dei corsi:
   (a) `USE DATABASE GestioneCorsiN`
   (b) `DROP TABLE Corsi`
   `DROP TABLE Docenti`
   • Per i gruppi che non hanno creato, durante la precedente esercitazione, il database per la gestione dei corsi:
   (a) `CREATE DATABASE GestioneCorsiN`
   (b) `USE DATABASE GestioneCorsiN`
   Questi due comandi vanno eseguiti separatamente.

2. I due comandi seguenti sono equivalenti:

```
CREATE TABLE Professori(
id decimal(5,0) PRIMARY KEY,
Cognome varchar(30) NOT NULL,
Nome varchar(30) NOT NULL );

CREATE TABLE Professori(
id decimal(5,0) [NOT NULL],
Cognome varchar(30) NOT NULL,
Nome varchar(30) NOT NULL
PRIMARY KEY(id) );
```

3. (a)
```
ALTER TABLE Professori
ADD Stipendio decimal(8,2)
DEFAULT 15000 CHECK (Stipendio >=0),
InCongedo bit DEFAULT 0;
```

(c)
```
ALTER TABLE Professori
ALTER COLUMN Stipendio decimal(9,2);
```
Il messaggio di SQL Server "`(2 row(s) affected)`" segnala quante tuple vengono modificate in seguito alla modifica della struttura della tabella `Professori`.

(d)
```
ALTER TABLE Professori
ALTER COLUMN Stipendio decimal(7,2);
```
Il messaggio di SQL Server "`Arithmetic overflow error converting numeric to data type numeric.`" segnala che la modifica della struttura della tabella porterebbe al troncamento di valori numerici già inseriti, e quindi non viene permessa.

(f)
```
ALTER TABLE Professori
DROP COLUMN InCongedo;
```
Il comando non va a buon fine, come segnalato dal messaggio d'errore "`ALTER TABLE DROP COLUMN InCongedo failed because one or more objects access this column.`" per la presenza di un vincolo di Default associato alla colonna `InCongedo` ("`The object 'DF__Professor__InCon__1CF15040' is dependent on column 'InCongedo'`" o simile).

(g) Il vincolo di Default può essere tolto, oltre che dall'Enterprise manager, anche con il comando T-SQL (il nome del constraint va sostituito con quello assegnato da SQL Server al proprio database):

```
ALTER TABLE Professori
DROP CONSTRAINT DF__Professor__InCon__1CF15040;
```

Dopodiché il comando al punto (f) va a buon fine.

4. I due comandi seguenti sono equivalenti:

```
CREATE TABLE Corsi(
id char(10) PRIMARY KEY,
CorsoDiLaurea varchar(30) NOT NULL,
Denominazione varchar(30) NOT NULL,
Professore decimal(5,0) FOREIGN KEY REFERENCES Professori (id)
[ ON DELETE NO ACTION ] ON UPDATE CASCADE,
Attivato bit DEFAULT 0
UNIQUE(CorsoDiLaurea, Denominazione)
);
```

```
CREATE TABLE Corsi(
id char(10) PRIMARY KEY,
CorsoDiLaurea varchar(30) NOT NULL,
Denominazione varchar(30) NOT NULL,
Professore decimal(5,0),
Attivato bit DEFAULT 0
UNIQUE(CorsoDiLaurea, Denominazione),
FOREIGN KEY (Professore) REFERENCES Professori (id)
[ON DELETE NO ACTION] ON UPDATE CASCADE
);
```

`[ON DELETE NO ACTION]` è opzionale perché `NO ACTION` è l'argomento di default (**Attenzione!** `NO ACTION` ha la semantica di RESTRICT in SQL:1999).

5.
```
CREATE TABLE Studenti(
Matricola varchar(10) PRIMARY KEY,
Cognome varchar(30) NOT NULL,
Nome varchar(30) NOT NULL,
CorsoDiLaurea varchar(30) NOT NULL,
Iscrizione char(9) NOT NULL,
Relatore decimal(5,0) FOREIGN KEY
REFERENCES Professori (id) ON UPDATE CASCADE);
```

6. (a) `SELECT * FROM Studenti;`

   (b) `SELECT * FROM Corsi WHERE Attivato=1;`

   (c) `SELECT Cognome, Nome, Iscrizione FROM Studenti WHERE CorsoDiLaura = 'Informatica' ORDER BY Cognome, Nome;`

   (d) `SELECT * FROM Studenti WHERE CorsoDiLaurea='Informatica' AND Relatore IS NULL;`
   Si noti che il Query Analyzer è case insensitive, quindi 'Informatica' è equivalente a 'INFORMATICA', etc. La query precedente è equivalente a
   `SELECT * FROM Studenti WHERE CorsoDiLaurea = 'INFORMATICA' AND Relatore IS NULL;`

   (e) `SELECT DISTINCT Denominazione FROM Corsi ORDER BY Denominazione DESC;`

   (f)
   ```
   SELECT Matricola, Cognome, Nome
   FROM Studenti
   WHERE CorsoDiLaurea = 'informatica' OR CorsoDiLaurea = 'matematica';
   ```

   (g)
   ```
   SELECT Matricola, CorsoDiLaurea FROM Studenti
   WHERE (CorsoDiLaurea = 'Fisica' AND Iscrizione = '1998/1999') OR Cognome LIKE 'P%'
   ```

(h)
```
SELECT Cognome, Nome
FROM Studenti
WHERE CorsoDiLaurea = ‘Informatica’
AND Iscrizione > ‘1999/2000’ AND (nome BETWEEN ‘Beatrice’ AND ‘Giovanni’);
```

(i)
```
SELECT Professori.Cognome, Professori.Nome, id, Stipendio
FROM Professori, Studenti
WHERE relatore = id AND CorsoDiLaurea LIKE ‘Ingegneria%’
AND (Stipendio > 10000 OR Stipendio < 15000);

SELECT Professori.Cognome, Professori.Nome, id, Stipendio
FROM Professori JOIN Corsi ON relatore = id
WHERE CorsoDiLaurea LIKE ‘Ingegneria%’
AND (Stipendio NOT BETWEEN 10000 AND 15000);
```

(j)
```
SELECT Cognome, Nome, Denominazione
FROM Professori,Corsi WHERE corsi.professore = professori.id
AND (CorsoDiLaurea = ‘ingegneria elettronica’ OR CorsoDILaurea = ‘informatica’)
AND Attivato = 1
ORDER BY Cognome, Nome;

SELECT Cognome, Nome, Denominazione
FROM Professori join Corsi on corsi.professore=professori.id
WHERE (CorsoDiLaurea = ’ingegneria elettronica’
OR CorsoDILaurea = ’informatica’)
AND Attivato = 1
ORDER BY Cognome, Nome;
```

(k)
```
SELECT Denominazione, Corsi.id
FROM Corsi, Professori, Studenti
WHERE Corsi.Professore=Professori.id and Studenti.Relatore=Professori.id
AND Studenti.CorsoDiLaurea = ‘informatica’
ORDER BY Corsi.id;

SELECT Denominazione, Corsi.id
FROM Corsi JOIN Professori ON Corsi.Professore = Professori.id JOIN Studenti
ON Studenti.Relatore=Professori.id
WHERE Studenti.CorsoDiLaurea = ‘informatica’ ORDER BY Corsi.id;
```