Basi di Dati:
Corso di laboratorio

Lezione 2

Raffaella Gentilini

Sommario

Soppressione di uno schema (DROP SCHEMA)

Per cancellare uno schema:

DROP SCHEMA: Sintassi

DROP SCHEMA <*nome_tabella*> [{CASCADE | RESTRICT}]

I parametri opzionali CASCADE e RESTRICT

- RESTRICT: Lo schema viene eliminato solo se vuoto (default), ovvero se non contiene alcuna definizione di tabelle, domini o altri oggetti;
- CASCADE: Vengono cancellati sia i dati che lo schema riferiti.

Soppressione di una tabella (DROP TABLE)

Per cancellare una tabella:

DROP TABLE: Sintassi

DROP TABLE <*nome_tabella*> [{CASCADE | RESTRICT}]

I parametri opzionali CASCADE e RESTRICT

- CASCADE: Gli oggetti che dipendono dalla tabella (vincoli, viste . . .) vengono cancellati automaticamente
- RESTRICT: La tabella non puo' venire cancellata se esistono oggetti dipendenti nella base di dati (default)

Esempi (I)

Si consideri lo schema relazionale visto nella lezione precedente:

- *persona*(*id_persona*, *codice_fiscale*, *nome*, *cognome*, *data_nascita*)
- corso(*id_corso*, *id_insegnante*, *sigla*, *crediti*, *descrizione*)
- frequenza(*id_studente*,*id_corso*,*voto*): dove *id_studente* ed *id_corso* sono chiavi esterne su persona e corso

Example

Il comando:

```
DROP TABLE corso;
```

porta ad un errore perche' la tabella *frequenza* dipende dalla tabella *corso*!

Esempi (II)

Example

Il comando:

```
DROP TABLE corso CASCADE;
```

- porta all'effettiva cancellazione della tabella *corso*;
- la tabella *frequenza* non viene cancellata;
- vengono invece soppressi i vincoli di chiave esterna su *corso* in *frequenza*.

Soppressione di un dominio (DROP DOMAIN)

Per cancellare un dominio:

DROP DOMAIN: Sintassi

DROP DOMAIN <*nome_dominio*> [{CASCADE | RESTRICT}]

I parametri opzionali CASCADE e RESTRICT

- RESTRICT: Il dominio viene eliminato solo se non e' utilizzato nella definizione di alcun altro oggetto dello schema logico (default);
- CASCADE: Viene eliminato il dominio e vengono modificate le definizioni di ogni colonna che lo utilizzi:
  - il tipo della colonna diventa quello espresso dalla definizione del dominio;
  - alla colonna viene dato l'eventuale valore di default del dominio;
  - i vincoli di integrita' del dominio diventano vincoli di colonna.

Modifica di una Tabella (ALTER TABLE)

ALTER TABLE: Sintassi

```
ALTER TABLE <nome_tabella> {
   ADD [COLUMN] <nome_col><dominio> [<vincolo_col>[...]] |
   DROP [COLUMN] <nome_col> [{CASCADE | RESRICT}] |
   ALTER [COLUMN] <nome_col> SET DEFAULT <valore> |
   ALTER [COLUMN] <nome_col> DROP DEFAULT |
   ADD CONSTRAINT <vincolo_tab> |
   DROP CONSTRAINT <nome_tab> [{CASCADE | RESRICT}] }
```

ALTER TABLE: Aggiunta di Colonne

Per aggiungere un nuovo attributo ad una tabella:

```
ALTER TABLE <nome_tabella>
  ADD [COLUMN] <nome_col><dominio> [<vincolo_col>[...]]
```

Example

- Aggiungere l'attributo *sesso* alla tabella *persona*

```
ALTER TABLE persona ADD COLUMN sesso CHAR;
```

- Se si aggiunge un attributo con vincolo `NOT NULL` bisogna prevedere un valore di default, che il sistema aggiungera' automaticamente a tutte le tuple gia' presenti:

```
ALTER TABLE persona
   ADD COLUMN istruzione CHAR(10)
   NOT NULL DEFAULT 'diploma';
```

ALTER TABLE: Soppressione di Colonne

Per sopprimere un attributo da una tabella:

ALTER TABLE <*nome_tabella*>
  DROP [COLUMN] <*nome_col*> [{CASCADE | RESRICT}]

Example

- Rimuovere l'attributo *sesso* alla tabella *persona*

  ALTER TABLE *persona* DROP COLUMN *sesso*;

- I parametri opzionali RESTRICT (default) e CASCADE indicano come comportarsi in caso di dipendenze

ALTER TABLE: Modifica di Valori di Default

Per modificare il valore di default di un attributo:

```
ALTER TABLE <nome_tabella>
  {ALTER [COLUMN] <nome_col> SET DEFAULT <valore> |
  ALTER [COLUMN] <nome_col> DROP DEFAULT}
```

Example

- Per l'attributo *sesso* inserire come valore di default 'F'

```
ALTER TABLE persona ALTER COLUMN sesso
SET DEFAULT 'F';
```

ALTER TABLE: Aggiunta di Vincoli

Per aggiungere un nuovo vincolo:

ALTER TABLE <*nome_tabella*> ADD CONSTRAINT <*def_vincolo*>

Example

ALTER TABLE *corso*
ADD CONSTRAINT *creditoMax* CHECK (*crediti* $\leq$ 12);

ALTER TABLE: Cancellazione di Vincoli

Sopprimere dei vincoli:

Example

```
ALTER TABLE corso DROP CONSTRAINT creditoMax;
```

Example

```
ALTER TABLE persona DROP UNIQUE (nome, cognome);
```

ALTER TABLE: Estensioni PostgreSQL

Rinomina di colonne

La rinomina di colonne e' un'estensione di PostgreSQL allo standard SQL

`ALTER TABLE` <*nome_tab*> `RENAME COLUMN` <*nome_1*> `TO` <*nome_2*>

Rinomina di tabella

La rinomina di tabelle e' un'estensione di PostgreSQL allo standard SQL

`ALTER TABLE` <*nome_tab*> `RENAME TO` <*nome*>

Modifica di un Dominio (ALTER DOMAIN)

ALTER DOMAIN: Sintassi

```
ALTER DOMAIN <nome_dominio> {
   SET <clausola_default> |
   DROP DEFAULT |
   ADD <vincolo_dominio> |
   DROP <vincolo_dominio>}
```

Data Manipulation Language (DML)

Istruzioni del DML

Le istruzioni del DML di SQL sono:

- Modifica
  - `INSERT`: Inserisce nuove tuple nel DB
  - `UPDATE` : Cancella tuple dal DB
  - `DELETE` : Modifica tuple nel DB
- Interrogazione
  - `SELECT`: Esegue interrogazioni (query) sul DB

INSERT INTO

INSERT INTO: Sintassi

INSERT INTO <*nome_tab*> [(<*nome_col*> [, . . .])]
VALUES [(<*valori*> [, . . .])]

- La lista dei valori [(<*valori*> [, . . .])] deve corrispondere con la lista delle colonne <*nome_tab*> [(<*nome_col*> [, . . .])]
- La lista degli attributi si puo' omettere, nel qual caso vale l'ordine con cui sono stati definiti
- le parole chiave DEFAULT e NULL possono prendere il posto di <*valori*>
- Se la lista non include tutti gli attributi, i restanti assumono valore NULL o il valore di default (se specificato)

Inserimento di Tuple

Example

Si consideri ad esempio l'inserimento di dati in una tabella con schema: *prodotto*(*cod*,*nome*,*prezzo*)

- I nomi degli attibuti possono essere omessi se si conosce l'ordine. Ad esempio, le due espressioni sotto equivalenti:

  `INSERT INTO` *prodotto(cod,prezzo,nome)* `VALUES` (123,3.40,'pc');
  `INSERT INTO` *prodotto* `VALUES` (123,'pc',3.40);

- Se alcuni valori sono nulli, possono essere omessi. Ad esempio, le due espressioni sotto equivalenti:

  `INSERT INTO` *prodotto* `VALUES` (123,3.40);
  `INSERT INTO` *prodotto* `VALUES` (123,3.40, `NULL`);

Inserimento di Tuple via Queries

E' anche possibile inserire in una tabella delle tuple che provengono dal risultato di una query. La sintassi e' la seguente:

```
INSERT INTO <nome_tab> [ (<nome_col> [,...]) ]
VALUES <select_query>
```

Tratteremo piu' avanti la scrittura di una query mediante il costrutto SELECT del DML di SQL.

Cancellazione di Tuple

DELETE: Sintassi

`DELETE FROM` <*nome_tab*> [`WHERE` <*predicato*>]

- L'istruzione `DELETE` puo' far uso di una condizione per specificare le tuple da cancellare;
- Tutte le tuple per cui il predicato <*predicato*> ha valore *vero* vengono cancellate;
- In assenza della clausola `WHERE`, vengono eliminate tutte le tuple della tabella a cui si fa riferimento

Cancellazione di Tuple

Example

1. Eliminare dall'iscrizione ad un determinato corso, gli studenti che non hanno ottenuto un voto pari almeno a 10:
   `DELETE FROM` *frequenza*
   `WHERE` *voto* `IS NOT NULL AND` *voto* < 10
2. Eliminare i corsi con meno di tre crediti:
   `DELETE FROM` *corso* `WHERE` *crediti* < 3

- Che succede se la cancellazione porta a violare il vincolo di integrita' referenziale?
- Ad esempio, che succede agli studenti che frequentavano i corsi con meno di tre crediti?
- ... lo vediamo tra posco ...

Aggiornamento di Tuple

UPDATE: Sintassi

`UPDATE` <*nome_tab*>
`SET` <*nome_col*> = { <*espressione*> | <*select_query*> } [, . . .]
`WHERE` <*predicato*>

- I valori delle colonne specificate sono modificati in tutte le tuple che soddisfano il predicato <*predicato*>
- <*espressione*> puo' far riferimento ai valori attuali delle tuple in via di modifica
- le parole chiave `NULL` e `DEFAAULT` costituiscono espressioni valide

Aggiornameno di Tuple

Example

- Aggiungere un punto a tutti i voti nella tabella *frequenza*

  `UPDATE` *frequenza*
  `SET` *voto* $=$ *voto* $+ 1$ `WHERE` *voto* `IS NOT NULL`

- Anche l'update puo' portare a violare vincoli di integrita referenziale

Politiche di Reazione

Violazione dei Vincoli e politiche di Reazione

Anziche' lasciare l programmatore il compito di garantire che a fronte di cancellazioni e modifiche della BD i vincoli di integrita' referenziale siano rispettati:

- si possono specificare opportune politiche di reazione in fase di definizione degli schemi

Politiche di Reazione

Example

- Tabella interna *impiegato*:

matricola	nome	cognome	dipartimento
A0001	Romolo	Neri	Acquisti
A0002	Remo	Bianchi	Vendite

- Tabella esterna *diparimento*

nome_dipartimento	sede	telefono
Acquisti	Roma	NULL
Vendite	Perugia	075558767

Violazioni operando sulla tabella interna

- Si possono introdurre violazioni modificando il contenuto della tabella interna in due modi:
  1. modificando il valore dell'attributo referente
  2. inserendo una nuova tupla
- per queste operazioni SQL non offre alcun supporto:
  - le operazioni vengono semplicemente impedite

Esempio

Example

- Tentativo di inserimento nella tabella interna che causa violazione:

matricola	nome	cognome	dipartimento
A0001	Romolo	Neri	Acquisti
A0002	Remo	Bianchi	Vendite
A003	Caligola	Verdi	Marketing

nome_dipartimento	sede	telefono
Acquisti	Roma	NULL
Vendite	Perugia	075558767

- L'inserimento viene impedito

Violare i vincoli operando sulla tabella esterna

- Diverse alternative per rispondere a violazioni generate da modifiche sulla tabella esterna (o tabella master)
- La tabella interna (o slave) deve adeguarsi alle modifiche che avvengono nella tabella master
- Le violazioni possono avvenire per:
  1. Modifiche dell'attributo riferito
  2. Cancellazione di tuple nella tabella master

Politiche di reazione per modifica attributo riferito

Cascade

La politica cascade prevede che in caso di modifica dell'attributo riferito:

- il nuovo valore dell'attributo della tabella esterna viene riportato su tutte le corrispondenti righe della tabella interna

Esempio (I)

Example

- modifica di un valore dell'attributo *nome_dipartimento* nella tabella esterna:

nome_dipartimento	sede	telefono
Acquisti	Roma	NULL
Vendite	Perugia	075558767

nome_dipartimento	sede	telefono
Acquisti	Roma	NULL
VenditeSuccursale	Perugia	075558767

Esempio (II)

Example

- Politica di reazione cascade:

matricola	nome	cognome	dipartimento
A0001	Romolo	Neri	Acquisti
A0002	Remo	Bianchi	Vendite

matricola	nome	cognome	dipartimento
A0001	Romolo	Neri	Acquisti
A0002	Remo	Bianchi	VenditeSuccursale

Politiche di reazione per modifica attributo riferito

Set Null

La politica set null prevede che in caso di modifica dell'attributo riferito:

- all'attributo referente (nella tabella interna) venga assegnato valore nullo al posto del valore modificato nella tabella esterna

Esempio (I)

Example

- modifica di un valore dell'attributo *nome_dipartimento* nella tabella esterna:

nome_dipartimento	sede	telefono
Acquisti	Roma	NULL
Vendite	Perugia	075558767

nome_dipartimento	sede	telefono
Acquisti	Roma	NULL
VenditeSuccursale	Perugia	075558767

Esempio (II)

Example

- Politica di reazione set null:

matricola	nome	cognome	dipartimento
A0001	Romolo	Neri	Acquisti
A0002	Remo	Bianchi	Vendite

⇓⇓⇓⇓⇓⇓⇓⇓⇓

matricola	nome	cognome	dipartimento
A0001	Romolo	Neri	Acquisti
A0002	Remo	Bianchi	NULL

Politiche di reazione per modifica attributo riferito

Set Default

La politica set default prevede che in caso di modifica dell'attributo riferito:

- all'attributo referente (nella tabella interna) venga assegnato un valore di default al posto del valore modificato nella tabella esterna

Esempio (I)

Example

- modifica di un valore dell'attributo *nome_dipartimento* nella tabella esterna:

nome_dipartimento	sede	telefono
Acquisti	Roma	NULL
Vendite	Perugia	075558767

nome_dipartimento	sede	telefono
Acquisti	Roma	NULL
VenditeSuccursale	Perugia	075558767

Esempio (II)

Example

- Politica di reazione set default:

matricola	nome	cognome	dipartimento
A0001	Romolo	Neri	Acquisti
A0002	Remo	Bianchi	Vendite

matricola	nome	cognome	dipartimento
A0001	Romolo	Neri	Acquisti
A0002	Remo	Bianchi	TBA

Politiche di reazione per modifica attributo riferito

No Action

La politica no action prevede che in caso di modifica dell'attributo riferito:

- Il sistema non innesca alcuna reazione speciale.
- Questo e' il valore di default e corrisponde semplicemente a rifiutare modifiche sui dati che portano alla violazione dell'integrita' refernziale.

Politiche di reazione per cancellazione tupla tabella esterna

- SQL mette a disposizione le stesse politiche di reazione:
  1. cascade: tutte le righe della tabella interna corrispondenti alla riga cancellata vengono cancellate
  2. set null: all'attributo referente viene assegnato il valore nullo al posto del valore presente nella riga cancellata dalla tabella esterna
  3. set default: all'attributo referente viene assegnato un valore di default al posto del valore presente nella riga cancellata dalla tabella esterna
  4. no action: il sistema non innesca alcuna azione speciale. Questo e' il valore di default e corrisponde semplicemente a rifiutare cancellazioni sui dati che violino l'integrita' referenziale.

Definizione dei vincoli interelazionali e specifica delle politiche reazione

La specifica della politica di reazione da adottare si pone nella definizione del vincolo di integrita':

- in una clausola di tipo `ON DELETE` (per le cancellazioni), oppure
- in una clausola di tipo `ON UPDATE` (per gli aggiornamenti).

Vincoli Interelazionali di Colonna: Sintassi

`REFERENCES` <*nome_tab*> [<*nome_col*>]
[{`ON DELTE` | `ON UPDATE`}
{ `NO ACTION` | `CASCADE` | `SET NULL` | `SET DEFAULT`}]

Definizione dei vincoli interelazionali e specifica delle politiche reazione

Vincoli Interelazionali di Tabella: Sintassi

`FOREIGN KEY` (<*nome_col*> [, . . .])
`REFERENCES` <*nome_tab*> (<*nome_col*> [, . . .])
[{`ON DELTE` | `ON UPDATE`}
{ `NO ACTION` | `CASCADE` | `SET NULL` | `SET DEFAULT`}]

Sommario

A titolo di ricapitolazione, vediamo come definire in PostgreSQL uno script per la generazione della BD corrispondente allo schema relazionale:

- `persona` (`id_persona`, `codice_fiscale`, `nome`, `cognome`, `data_nascita`)
- `corso` (`id_corso`, `id_insegnante`, `sigla`, `crediti`, `descrizione`)
- `frequenza` (`id_studente, id_corso`, `voto`)

introdotto nell'esercitazione precedente. In particolare:

- il nostro script prevedera' la cancellazione preventiva dalla base di dati le tabelle *persona,corso,frequenza*, se presenti;
- si definiranno opportune politiche di reazione alla modifica/cancellazione dei dati.

Definizione di Script

```
DROP TABLE persona;
DROP TABLE corso;
DROP TABLE frequenza;
CREATE TABLE persona (
id_persona INTEGER PRIMARY KEY,
codice_fiscale CHAR(11) UNIQUE,
nome VARCHAR(40) NOT NULL,
cognome VARCHAR(40) NOT NULL,
data_nascita DATE);
CREATE TABLE corso (
id_corso INTEGER PRIMARY KEY,
id_insegnante INTEGER REFERENCES persona (id_persona) ON DELETE
SET NULL,
sigla CHAR(7) UNIQUE NOT NULL,
crediti INTEGER CHECK(crediti > 0 OR crediti IS NULL),
descrizione TEXT);
```

Definizione di Script

```
CREATE TABLE frequenza (
id_studente INTEGER REFERENCES persona ON DELETE CASCADE,
id_corso INTEGER REFERENCES corso ON DELETE CASCADE,
voto INTEGER CHECK ( voto > 0 AND voto ≤ 30 ),
PRIMARY KEY (id_studente,id_corso));
```

Bibliografia

Bibliografia ed Approfondimenti

- R.A.Elmasri, S.B. Navathe. Sistemi di Basi di Dati – Fondamenti: Capitolo 8 (8.2).
- Capitolo 5 (Data Definition) e Capitolo 6 (Data Manipulation) del manuale di PostgreSQL (http://www.postgresql.org/docs/manuals/)