



UNIVERSITÀ DEGLI STUDI DI SALERNO

Atzeni, Ceri, Fraternali, Paraboschi, Torlone
Basi di dati Quinta edizione
 McGraw-Hill Education, 2018

1

BASI DI DATI

Concetti Base SQL-DML - prima parte



Matteo Gaeta

Full Professor – Senior Member IEEE

Le categorie di oggetti presenti nel mondo fisico non hanno criteri di appartenenza definiti con precisione. (Lotfi A. Zadeh)

BASI DI DATI

Concetti base – SQL

2

OPERAZIONI SUI DATI

➤ **Interrogazione:**

✓ **SELECT**

➤ **modifica:**

✓ **INSERT, DELETE, UPDATE** (Operazioni di Aggiornamento)

➤ **operazioni di**

✓ **inserimento: INSERT**

✓ **eliminazione: DELETE**

✓ **modifica: UPDATE**

OPERAZIONI DI AGGIORNAMENTO

- Possono essere effettuare su una o più ennuple di una relazione, nonché sulla base di una condizione che può coinvolgere anche altre relazioni

L'algebra e il denaro sono essenzialmente livellatori; la prima intellettualmente, l'altro effettivamente. (Simone Weil)

BASI DI DATI

Concetti base – SQL

3

INSERIMENTO DI ENNUPLE

➤ **INSERT INTO** Tabella [(Attributi)]
VALUES(Valori)

oppure

➤ **INSERT INTO** Tabella [(Attributi)]
SELECT ...

❖ **INSERT INTO** Persone **VALUES**('Mario',25,52);

❖ **INSERT INTO** Persone(Nome, Eta, Reddito) **VALUES**('Pino',25,52);

❖ **INSERT INTO** Persone(Nome, Reddito) **VALUES**('Lino',55);

❖ **INSERT INTO** Persone (Nome)

SELECT Padre

FROM Paternita

WHERE Padre NOT IN (SELECT Nome

FROM Persone);

Man mano che la complessità aumenta, affermazioni precise perdono significato e affermazioni significative perdono precisione. (Lotfi A. Zadeh)

BASI DI DATI

Concetti base – SQL

4

INSERIMENTO DI ENNUPLE

➤ per ogni attributo specificare il valore corrispondente del giusto dominio, l'ordinamento degli attributi (se presente) e dei valori è significativo

➤ le due liste debbono avere lo stesso numero di elementi,

✓ se la lista di attributi è omessa, si fa riferimento a tutti gli attributi della relazione, secondo l'ordine con cui sono stati definiti

✓ se la lista di attributi non contiene tutti gli attributi della relazione, per gli altri viene inserito un NULL (se permesso) o il valore di default

❖ **ATTENZIONE: ricordare che una tabella in SQL è definita come un multinsieme di tuple. Ciò significa che potranno essere inserire tuple identiche in quanto una tabella SQL non è una relazione. E' quindi necessario inserire i vincoli di chiave.**

BASI DI DATI

Concetti base – SQL

5

INSERIMENTO DI ENNUPLE: ESEMPI

```
create table impiegato (
  matricola char(4) primary key,
  cognome varchar(20) not null);

insert into impiegato values('AAA','Pippo');

select * from impiegato;
```

Riquadro di «output»

	matricola character(4)	cognome character varying(20)
1	AAA	Pippo

Unix Ln 5, Col 31, Ch 121 1 row.

BASI DI DATI

Concetti base – SQL

6

INSERIMENTO DI ENNUPLE: ESEMPI

```
create table impiegato (
  matricola char(4) primary key,
  cognome varchar(20) not null,
  constraint MatrLen check(char_length(matricola)=4));

insert into impiegato values('AAA','Pippo');
```

Riquadro di «output»

	Output dei dati	Explain	Messaggi	Cronologia
ERROR: new row for relation "impiegato" violates check constraint "matrlen" DETAIL: Failing row contains (AAA , Pippo).				

ERROR: new row Unix Ln 1, Col 33, Ch 33 13 ms

BASI DI DATI

Concetti base – SQL

7

INSERIMENTO DI ENNUPLE: ESEMPI

```
Query - UNISA on postgres@localhost:5432 *
```

SQL Editor Graphical Query Builder

Previous queries Delete Delete All

```
insert into impiegato(matricola,cognome) values('AAAA','Rossi');
insert into impiegato(matricola,cognome) values('BBBB','Bianchi');
insert into impiegato(matricola,cognome) values('BBBB','Neri');
insert into impiegato(matricola,cognome) values('DDDD','Verdi');
```

Output pane

Data Output Explain Messages History

```
[QUERY ] insert into impiegato(matricola,cognome) values('AAAA','Rossi')
[QUERY ] insert into impiegato(matricola,cognome) values('BBBB','Bianchi')
[WARNING] insert into impiegato(matricola,cognome) values('BBBB','Neri')
          ERROR: duplicate key value violates unique constraint "impiegato_pkey"
          DETAIL: Key (matricola)=(BBBB) already exists.
[QUERY ] insert into impiegato(matricola,cognome) values('DDDD','Verdi')
```

BASI DI DATI

Concetti base – SQL

8

INSERIMENTO DI ENNUPLE: ESEMPI

```
insert into impiegato(matricola,cognome) values('AAAA','Rossi');
insert into impiegato(matricola,cognome) values('BBBB','Bianchi');
insert into impiegato(matricola,cognome) values('BBBB','Neri');
insert into impiegato(matricola,cognome) values('DDDD','Verdi');
```

❖ La Terza ennupla non viene inserita in quanto viola il vincolo di primary key sull'attributo matricola il cui valore «BBBB» esiste già nella ennupla relativa a Bianchi, precedentemente inserita.

```
Query - UNISA on postgres@localhost:5432 *
```

SQL Editor Graphical Query Builder

Previous queries Delete Delete All

```
select * from impiegato
```

Output pane

Data Output Explain Messages History

	matricola character(4)	cognome character varying(20)
1	AAAA	Rossi
2	BBBB	Bianchi
3	DDDD	Verdi

OK. Unix Ln 1, Col 24, Ch 24 3 rows. 17 ms

BASI DI DATI

Concetti base – SQL

9

ELIMINAZIONE DI ENNUPLE

➤ **DELETE FROM Tabella**

[WHERE Condizione]

❖ elimina le ennuple che soddisfano la condizione

➤ **N.B. ovviamente vengono eliminate le righe (ennuple, occorrenze, tuple, ecc.) e non lo schema della Tabella.**

❖ Esempio Tabella Dipartimento(Codice:char(4),Nome:varchar(20));

DELETE from Dipartimento

WHERE Nome is NULL;

➤ Se la condizione where manca allora la tabella viene svuotata.

➤ **DELETE from Dipartimento**, elimina il contenuto della tabella e non lo schema

➤ **DROP table Dipartimento**, elimina la tabella ovvero elimina lo schema

BASI DI DATI

Concetti base – SQL

10

ELIMINAZIONE DI ENNUPLE: ESEMPI

➤ **DELETE FROM Persone**

WHERE Eta < 35

➤ **DELETE FROM Paternita**

WHERE Figlio NOT in

(SELECT Nome

FROM Persone)

➤ **DELETE FROM Paternita**

❖ **ATTENZIONE: DELETE** può causare eliminazioni da altre relazioni (se i vincoli di integrità referenziale sono definiti con politiche di reazione cascade) o altri effetti indesiderati

❖ **ATTENZIONE: se la WHERE viene omessa, si intende where «TRUE»**

BASI DI DATI

Concetti base – SQL

11

MODIFICA DI ENNUPLE

➤ UPDATE NomeTabella

SET Attributo = <ESPRESSIONE | NULL | DEFAULT>
{,Attributo = <ESPRESSIONE | NULL | DEFAULT>}
[WHERE Condizione]

- ❖ La clausola WHERE viene applicata per prima e determina quindi le ennuple da modificare
- ❖ La clausola SET ovviamente determina i nuovi valori dell'attributo coinvolto
- ❖ Esempio: Incrementare di 5 lo stipendio dei dipendenti afferenti al settore amministrazione
- ❖ **UPDATE Dipendente SET Stipendio = Stipendio+5**
WHERE settore='amministrazione';

BASI DI DATI

Concetti base – SQL

12

MODIFICA DI ENNUPLE: ESEMPI

➤ UPDATE Persone

SET Reddito = 45
WHERE Nome = 'Piero'

➤ UPDATE Persone

SET Reddito = Reddito * 1.1
WHERE Eta < 30

➤ UPDATE prodotto SET prezzo = prezzo * 1.10

Esempio tabella

Dipartimento(Codice:char(4),Nome:varchar(20))

- Update Dipartimento set
Nome = 'Amministrazione'
- Update Dipartimento Set
Nome = 'Amministrazione'
where Codice = 'aaaa'
- Update Dipartimento Set
Nome = 'Sconosciuto'
where Nome is Null
- Update Dipartimento Set
Nome = 'Amministrazione', Codice
= 'bbbb' where Codice = 'aaaa'

BASI DI DATI

Concetti base – SQL

13

TUTTO OPPURE NIENTE

- **Warning: Un comando SQL è eseguito con modalità tutto o niente**
- **CREATE** table impiegato (
 - matricola char(4) primary key check (char_length(matricola) = 4),
 - cognome varchar(20) not null,
 - nome varchar(20) not null,
 - stipendio numeric(8,2) not null check (stipendio >= 1000 and stipendio <= 10000));
- **insert into** impiegato(matricola, cognome, nome, stipendio)
 - values('AAAA', 'Paolino', 'Paperino', 8000);
 - insert into** impiegato(matricola, cognome, nome, stipendio)
 - values('BBBB', 'Parker', 'Ken', 5000);
 - insert into** impiegato(matricola, cognome, nome, stipendio)
 - values('CCCC', 'Willer', 'Tex', 4000);
 - insert into** impiegato(matricola, cognome, nome, stipendio)
 - values('DDDD', 'Totti', 'Francesco', 9000);

BASI DI DATI

Concetti base – SQL

14

TUTTO OPPURE NIENTE

CREATE table impiegato (

- matricola char(4)
 - primary key check (char_length(matricola) = 4),
- cognome varchar(20) not null,
- nome varchar(20) not null,
- stipendio numeric(8,2) not null check (stipendio >= 1000 and stipendio <= 10000));

Query - Prova on postgres@localhost:5432 *

SQL Editor Graphical Query Builder

Previous queries Delete Delete All

select * from impiegato;

Output pane

Data Output Explain Messages History

	matricola character(4)	cognome character varying(20)	nome character varying(20)	stipendio numeric(8,2)
1	AAAA	Paolino	Paperino	8000.00
2	BBBB	Parker	Ken	5000.00
3	CCCC	Willer	Tex	4000.00
4	DDDD	Totti	Francesco	9000.00

OK. Unix Ln 1, Col 25, Ch 25 4 rows. 19 ms

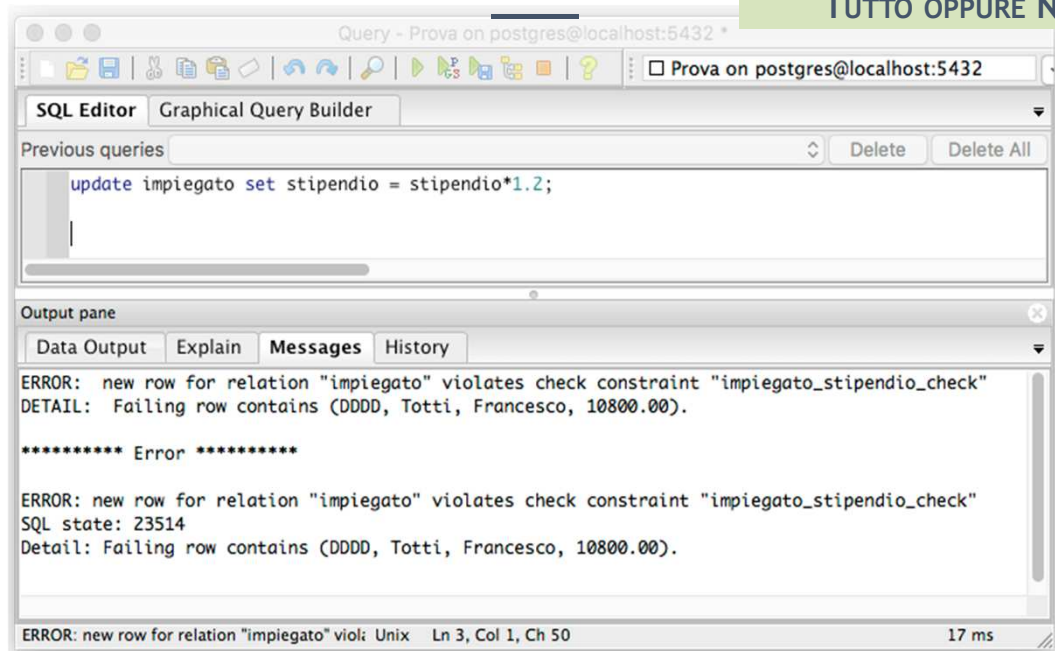
➤ **COSA SUCCEDDE CON IL COMANDO
UPDATE SET IMPIEGATO STIPENDIO=
STIPENDIO*1.2 ?**

BASI DI DATI

Concetti base – SQL

15

TUTTO OPPURE NIENTE



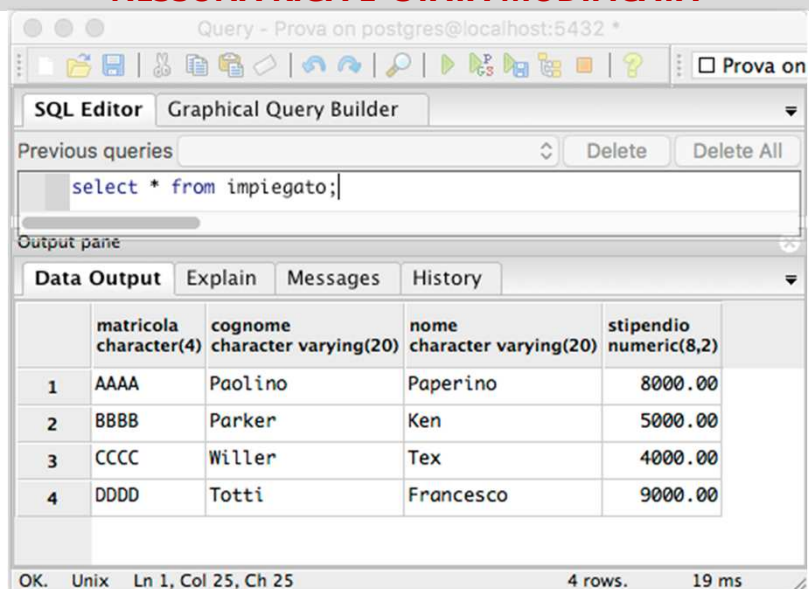
BASI DI DATI

Concetti base – SQL

16

TUTTO OPPURE NIENTE

NESSUNA RIGA E' STATA MODIFICATA



BASI DI DATI

17

Concetti base – SQL

GARANTIRE INTEGRITÀ REFERENZIALE

➤ **Esempio**, creare le seguenti tabelle in SQL:

- ❖ **PERSONE(nome, reddito, età, sesso)**: nome, stringa di 20 caratteri (chiave primaria); reddito, valore reale; età, intero di 3 cifre; sesso, un carattere;
- ❖ **GENITORE(figlio, genitore)**: figlio, stringa di 20 caratteri (chiave esterna su Persone) e valore di default 'Jack'; genitore, stringa di 20 caratteri (chiave esterna su persone); Chiave Primaria tabella formata da «figlio» e «genitore»;

```
CREATE TABLE PERSONE (
  nome VARCHAR(20) PRIMARY KEY,
  reddito REAL,
  eta NUMERIC(3),
  sesso CHAR
)
```

```
CREATE TABLE GENITORI (
  figlio VARCHAR(20) DEFAULT 'Jack'
  REFERENCES PERSONE,
  genitore VARCHAR(20)
  REFERENCES PERSONE,
  PRIMARY KEY (figlio, genitore)
)
```

BASI DI DATI

18

Concetti base – SQL

GARANTIRE INTEGRITÀ REFERENZIALE

➤ **OPZIONI in caso di cancellazione o modifica di una riga (Target) di PERSONE:**

1. Cancellare (aggiornare) tutte le righe di GENITORI che refenziano la riga Target (di PERSONE)
2. Non permettere la cancellazione (aggiornamento) della riga Target (di PERSONE)
3. Per ogni riga di GENITORI referenziata dalla riga Target (di PERSONE) cancellata (aggiornata) impostare i valori degli attributi figlio e genitore ai valori di default
 - ❖ Si osservi che questa opzione è in conflitto con il fatto che il valore di default per genitore sarebbe NULL, ma dato che genitore è parte della Primary Key di GENITORI tale valore non può essere impostato a NULL
4. Per ogni riga di GENITORI eliminata (aggiornata), impostare i corrispondenti valori degli attributi figlio e genitore referenzianti a NULL
 - ❖ Si osservi che tale opzione è in conflitto con il fatto che figlio e genitore sono parte di Primary Key di GENITORI, quindi non possono essere impostati a NULL

BASI DI DATI

19

Concetti base – SQL

LIMITAZIONI E POLICY

- Nell'ambito di una creazione di una Tabella (CREATE TABLE ...), data una FOREIGN KEY (attributi) REFERENCES (tabella referenziata), e le possibili opzioni ON DELETE e ON UPDATE le limitazioni utilizzabili sono:
- ❖ **NO ACTION** (RESTRICT): la cancellazione (modifica) della riga target non viene consentita perché essa viola i vincoli di FOREIGN KEY, in assenza di indicazione specifiche NO ACTION è il valore di default.
 - ❖ **CASCADE**: aggiorna i valori nella tabella referenziante partendo dal valore della tabella referenziata
 - ❖ **SET DEFAULT**: all'attributo referenziante viene assegnato il valore di DEFAULT al posto del valore modificato nella tabella referenziata
 - ❖ **SET NULL**: all'attributo referenziante viene assegnato il valore NULL al posto del valore modificato nella tabella referenziata

BASI DI DATI

20

Concetti base – SQL

ESEMPIO LIMITAZIONI E POLICY

➤ **CREATE TABLE GENITORI (**
figlio VARCHAR(20) DEFAULT 'Jack' REFERENCES PERSONE
ON UPDATE CASCADE ON DELETE SET DEFAULT,
genitore VARCHAR(20) REFERENCES PERSONE,
ON UPDATE NO ACTION ON DELETE NO ACTION
PRIMARY KEY (figlio, genitore))

➤ **DELETE FROM PERSONE WHERE Sesso='F', Cosa succede?**

PERSONE

Nome	Reddito	Età	Sesso
Mario	15	80	M
Carlo	25	24	M
Giuseppe	30	45	M
Maria	76	43	F
Gianni	60	50	M
Francesca	18	26	F

PERSONE

Nome	Reddito	Età	Sesso
Mario	15	80	M
Carlo	25	24	M
Giuseppe	30	45	M
Maria	76	43	F
Gianni	60	50	M

Figlio	Genitore
Carlo	Gianni
Carlo	Maria
Francesca	Giuseppe

GENITORI

Figlio	Genitore
Carlo	Gianni
Carlo	Maria
Jack	Giuseppe

- ✓ La riga 'Francesca' nella REFERENZIATA è eliminata, il corrispondente valore di figlio in GENITORI è modificato al valore Default (Jack) (figlio ...ON DELETE SET DEFAULT); la riga 'Maria' non è eliminata, violerebbe il vincolo Primary Key e la policy (genitore ... ON DELETE NO ACTION)

BASI DI DATI

21

➤ **CREATE TABLE GENITORI (** **Concetti base – SQL**
figlio VARCHAR(20) DEFAULT 'Jack' REFERENCES PERSONE
ON UPDATE CASCADE ON DELETE SET DEFAULT,
genitore VARCHAR(20) REFERENCES PERSONE,
ON UPDATE NO ACTION ON DELETE NO ACTION
PRIMARY KEY (figlio, genitore))

ESEMPIO LIMITAZIONI E POLICY

➤ **UPDATE PERSONE SET Nome='Andrea' WHERE Eta=24 Cosa succede?**

PERSONE

Nome	Reddito	Età	Sesso
Mario	15	80	M
Carlo	25	24	M
Giuseppe	30	45	M
Maria	76	43	F
Gianni	60	50	M

PERSONE

Nome	Reddito	Età	Sesso
Mario	15	80	M
Andrea	25	24	M
Giuseppe	30	45	M
Maria	76	43	F
Gianni	60	50	M

Figlio	Genitore
Carlo	Gianni
Carlo	Maria
Francesca	Giuseppe

GENITORI

Figlio	Genitore
Andrea	Gianni
Andrea	Maria
Jack	Giuseppe

✓ La riga 'Carlo' nella REFERENZIATA è variata in 'Andrea', il corrispondente valore figlio in GENITORI è variato in CASCADE ad 'Andrea' (figlio ...ON UPDATE CASCADE) e non viola il vincolo Primary Key; con più PERSONE di età=24 le modifiche non sarebbero effettuate, violando il vincolo Primary Key sull'attributo nome in caso di genitori con lo stesso valore

BASI DI DATI

22

Materiale utilizzato e bibliografia

➤ Le slide utilizzate dai docenti per le attività frontali sono in gran parte riconducibili e riprese dalle slide originali (con alcuni spunti parziali ripresi dai libri indicati) realizzate da:

✓ autori del libro Basi di Dati (Atzeni e altri) testo di riferimento del corso Basi di Dati e sono reperibili su internet su molteplici link oltre che laddove indicato dagli stessi autori del libro;

✓ Prof.ssa Tiziana Catarci e dal dott. Ing. Francesco Leotta – corso di Basi di Dati dell'Università degli Studi La Sapienza di Roma al seguente link ed altri: <http://www.dis.uniroma1.it/~catarci/basidatGEST.html> (molto Interessanti anche le lezioni su YouTube).

✓ Prof. Luca Allulli e Umberto Nanni, Libro Fondamenti di basi di dati, editore HOEPLI (testo di facile lettura ed efficace).

➤ Diverse slide su specifici argomenti utilizzate dai docenti per le attività frontali sono anche in parte riconducibili e riprese dalle slide originali – facilmente reperibili e accessibili su internet - realizzate da:

Prof.ssa Roberta Aiello – corso Basi di Dati dell'Università di Salerno

Prof. Dario Maio – corso Basi di Dati dell'Università di Bologna al seguente link ed altri: <http://bias.csr.unibo.it/maio>

Prof. Marco Di Felice – corso Basi di Dati dell'Università di Bologna al seguente link ed altri: <http://www.cs.unibo.it/difelice/dbsi/>

Prof. Marco Maggini e prof. Franco Scarselli – corso Basi di Dati dell'Università di Siena ai seguenti link ed altri: [http://staff.icar.cnr.it/pontieri/didattica/LabSI/lezioni/_preliminari-DB1%20\(Maggini\).pdf](http://staff.icar.cnr.it/pontieri/didattica/LabSI/lezioni/_preliminari-DB1%20(Maggini).pdf)

Prof.ssa Raffaella Gentilini – corso Basi di Dati dell'Università di Perugia al seguente link ed altri: <http://www.dmi.unipg.it/raffaella.gentilini/BD.htm>

Prof. Enrico Giunchiglia – corso Basi di Dati dell'Università di Genova al seguente link ed altri: <http://www.star.dist.unige.it/~enrico/BasiDiDati/>

Prof. Maurizio Lenzerini – corso Basi di Dati dell'Università degli Studi La Sapienza di Roma al seguente link ed altri: <http://didattica.info.altervista.org/Quinta/Database2.pdf>

➤ The PostgreSQL Global Development Group - PostgreSQL nn.xx Documentation

➤ PostgreSQL (appendice - scaricabile dal sito del libro (area studenti) e www.postgresql.org)