



UNIVERSITÀ DEGLI STUDI DI SALERNO

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





DIEM
DIPARTIMENTO DI
INGEGNERIA DELL'INFORMAZIONE ED ELETTRICA E MATEMATICA APPLICATA

Matteo Gaeta
Full Professor – Senior Member IEEE

Vincoli di integrità generici: check

- SQL consente di definire vincoli su colonne e tabelle.
- Un vincolo di controllo è il tipo di vincolo più generico. Esso consente di specificare che il valore in una determinata colonna deve soddisfare un'espressione booleana (valore di verità).
 - ✓ Ad esempio, per richiedere prezzi dei prodotti positivi, potresti utilizzare:


```
CREATE TABLE products (
  product_no integer,
  name text,
  price numeric CHECK (price > 0) );
```
- Un vincolo check è costituito dalla parola chiave CHECK seguita da un'espressione tra parentesi.
 - Specifica di vincoli di ennuola (e anche vincoli più complessi, non sempre supportati)


```
check ( Condizione )
```

Vincoli di integrità generici: check

```

1  /* Script Creazione di una Tabella con un Vincolo di Colonna */
2  DROP TABLE IF EXISTS PRODUCTS;
3  Create Table PRODUCTS(
4  Product_no Integer,
5  name text,
6  price numeric CHECK (price >0));
7
8  INSERT INTO PRODUCTS (Product_no, name, price)
9  VALUES (001, 'Tennent's', '1' );
10
11 Select *
12 FROM PRODUCTS;
13
14 /* Script Inserimento di una riga che viola il vincolo */
15 INSERT INTO PRODUCTS (Product_no, name, price)
16 VALUES (002, 'Tennent's', '-1' );
17
18
19

```

Query Editor Query History Scratch Pad

product_no	name	price
1	Tennent's	1

Data Output Explain Messages Notifications

ERROR: ERRORE: la nuova riga per la relazione "products" viola il vincolo di controllo "products_price_check"
 DETAIL: La riga in errore contiene (2, Tennent's, -1).

SQL state: 23514

Vincoli di integrità generici: check

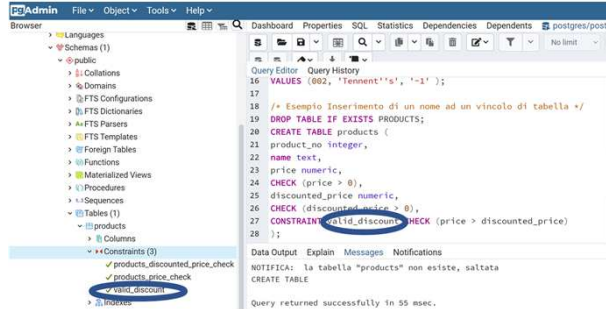
- L'espressione del vincolo di controllo dovrebbe coinvolgere la colonna così vincolata, altrimenti il vincolo non avrebbe molto senso.
- E' possibile dare al vincolo un nome separato, ciò è utile per i messaggi di errore e consente di fare riferimento al vincolo per modificarlo.
- Nel seguito un esempio che mostra tale sintassi è:


```

CREATE TABLE products (
product_no integer,
name text,
price numeric CONSTRAINT positive_price CHECK (price > 0) );

```
- Per specificare e denominare un vincolo, occorre utilizzare la parola chiave CONSTRAINT seguita da un identificatore seguito dalla definizione del vincolo.

Vincoli di integrità generici: check



- Un vincolo di controllo è soddisfatto se l'espressione di controllo restituisce true o il valore NULL.
- Warning: PostgreSQL non supporta i vincoli CHECK che fanno riferimento a dati della tabella diversi dalla riga nuova o aggiornata da controllare

Vincoli di integrità generici: check

- SQL permette di specificare vincoli di integrità generici sugli attributi e le tabelle attraverso la clausola check.
- Le condizioni utilizzabili sono le stesse che possono apparire come argomento della clausola where di un'interrogazione SQL.
- Vincoli di CHECK

**CREATE TABLE NomeTabella(
NomeAttributoDominio [check(Condizione)]**

Vincoli di CHECK - **ATTENZIONE**

- La condizione contenuta nel vincolo di *check* viene valutata **immediatamente dopo** l'inserimento/modifica di una tupla nella tabella in cui il vincolo è definito.

❖ *Esempio*

Impiegato

Nome	Cognome	Dipart	StipAnn
------	---------	--------	---------

check (Cognome like 'c%')

check (StipAnn >= 1 AND StipAnn <= 999)

```
INSERT INTO Impiegato(Nome,Cognome,Dipart,StipAnn)
VALUES ('Marco','Cesa','Produzione',800)
```

L'inserimento va a **buon fine**, perché soddisfa tutti i vincoli (compresi quelli di *check*) specificati per la tabella.

Vincoli di CHECK - **ATTENZIONE**

- In caso di violazione del vincolo, l'operazione di modifica/inserimento che ha causato tale violazione viene "disfatta" dal sistema (**Rollback parziale**).

❖ *Esempio*

Impiegato

Nome	Cognome	Dipart	StipAnn
------	---------	--------	---------

check (Cognome like 'c%')

check (StipAnn >= 1 AND StipAnn <= 999)

```
INSERT INTO Impiegato(Nome,Cognome,Dipart,StipAnn)
VALUES ('Andrea','Marrella','Amministrazione',800)
```

Inizialmente l'inserimento della tupla viene effettuato senza verificare i vincoli di *check* specificati per la tabella.

A questo punto si verifica se l'inserimento ha violato qualche vincolo. Dato che il valore 'Marrella' non inizia con la 'c', un vincolo di *check* è stato violato.

La tupla viene eliminata dalla tabella.

Rollback parziale

Nome	Cognome	Dipart	StipAnn
Andrea	Marrella	Amministrazione	800

Vincoli di integrità generici: check

- Il vincolo di check è un costrutto molto potente :
- la condizione specificata nel vincolo può far riferimento ad altri attributi della stessa tabella oppure ad attributi in tabelle differenti.
- La condizione contenuta nel vincolo di check deve essere sempre verificata affinché la base di dati sia corretta

Vincoli di integrità generici: check

- I vincoli predefiniti (PRIMARY KEY, UNIQUE, FOREIGN KEY, ...) possono essere espressi attraverso il vincolo di check.

```

CREATE TABLE Impiegato(
  Nome VARCHAR(20),
  Cognome VARCHAR(20),
  Dipart VARCHAR(20),
  Check
  -- NOT NULL
  (Nome is not null AND Cognome is not null AND
  -- UNIQUE e quindi PRIMARY KEY
  1 >= (SELECT count(*)
    FROM Impiegato I
    WHERE Nome = I.Nome and Cognome = I.Cognome ))
  )

```

Vincoli di integrità generici: check

- Quando un vincolo di check coinvolge due o più tabelle, possono emergere comportamenti anomali non desiderati.

```
CREATE TABLE Impiegato( Nome VARCHAR (20),
Cognome VARCHAR (20),
Dipart VARCHAR (20),
StipAnn INT,
check ((SELECT count(distinct Dipart) FROM Impiegato) +
(SELECT count(Nome) FROM Dipartimento) <100 ) );
```

- Se inseriamo nella tabella Dipartimento un numero di Dipartimenti > 100, il vincolo rimane soddisfatto, dato che il controllo viene effettuato solo quando inseriamo tuple dentro la tabella Impiegato...ciò non è soddisfacente.

Vincoli di integrità generici: asserzioni

- Per evitare eventuali anomalie comportate dai vincoli di check su più tabelle, si possono utilizzare le asserzioni.
- Le asserzioni rappresentano vincoli non associati ad attributi o tabella in particolare, ma appartengono direttamente allo schema della Base di Dati.
- Le asserzioni permettono di esprimere vincoli particolari, come vincoli su più tabelle o vincoli che richiedono una cardinalità min\max per una tabella.

```
CREATE ASSERTION NomeAsserzione
check(Condizione)
```

Vincoli di integrità generici: asserzioni

- Specifica vincoli a livello di schema

create assertion NomeAss check (Condizione)

**create assertion AlmenoUnImpiegato
check (1 <= (select count(*)
from Impiegato))**

- ❖ **Spesso non supportata**

Politiche di controllo nei vincoli di integrità

- ❖ La verifica dei vincoli può essere

- **immediate** (immediata): la loro violazione annulla l'ultima modifica

- **deferred** (differita): la loro violazione annulla l'intera transazione

Politiche di controllo nei vincoli di integrità

- Ogni vincolo di integrità, definito tramite check o tramite asserzione, è associato ad una politica di controllo che specifica se il vincolo è immediato o differito.
- I vincoli immediati (valore di default) – sono verificati dopo ogni operazione che coinvolge le tabelle presenti nel vincolo di check o nell'asserzione.
- Quando un vincolo immediato non è soddisfatto, l'operazione di modifica che ha causato la violazione è stata appena eseguita e il sistema può disfarla. Questo modo di procedere è chiamato Rollback Parziale

Politiche di controllo nei vincoli di integrità

- Tutti i vincoli predefiniti (PRIMARY KEY, UNIQUE, FOREIGN KEY, NOT NULL) sono verificati in modo immediato.
- I vincoli differiti sono verificati solo al termine dell'esecuzione di una serie di operazioni (che costituisce una transazione).
- Il controllo differito viene tipicamente introdotto per gestire situazioni in cui non è possibile costruire una situazione consistente con una singola modifica della base di dati.

Politiche di controllo nei vincoli di integrità

- Si pensi, ad esempio, ad una tabella Dipartimenti il cui attributo Direttore è associato ad un vincolo di integrità referenziale verso una tabella Impiegato, e la tabella Impiegato presenta a sua volta un attributo Dipart associato ad un vincolo di integrità referenziale verso la tabella Dipartimenti.
- Se entrambi i vincoli fossero immediati, non sarebbe possibile modificare lo stato iniziale vuoto delle due tabelle, in quanto ogni singolo comando di inserimento di tuple non rispetterebbe il vincolo di integrità referenziale.
- Il controllo differito permette di gestire agevolmente questa situazione.

Politiche di controllo nei vincoli di integrità

- Quando si verifica una violazione di un vincolo differito al termine della transazione, non c'è modo di individuare l'operazione che ha causato la violazione.
- Diventa perciò necessario disfare l'intera sequenza di operazioni che costituiscono la transazione. in questo caso si esegue un rollback.
- Grazie ai meccanismi di controllo immediato\deferito, l'esecuzione di un comando di modifica dell'istanza della base di dati che soddisfa tutti i vincoli, produrrà una nuova istanza consistente della base di dati (cioè, che soddisfa a sua volta tutti i vincoli).

BASI DI DATI

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).
- ✓ Proff. 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://didatticainfo.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)