



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-DDL – quinta parte



Matteo Gaeta

Full Professor – Senior Member IEEE

Non ci può essere inventiva, fantasia, creazione del nuovo se si comincia dal seppellire se stessi, la propria storia e la realtà. (Enrico Berlinguer)

BASI DI DATI

2

Concetti base – SQL

SCHEMI E VINCOLI DI INTEGRITÀ

- Un **VINCOLO DI INTEGRITÀ** descrive le condizioni che ogni istanza legale di una relazione deve soddisfare.
 - ✓ Un vincolo di integrità limita i dati che possono essere memorizzati in un'istanza della base dati
 - ✓ Insert/delete/update che violano i vincoli di integrità non sono permessi
 - ✓ I vincoli di integrità possono essere usati per garantire la semantica dell'applicazione o per prevenire incosistenze della base dati.
- **Vincoli di Integrità INTRARELAZIONALI:** coinvolgono una sola relazione (NOT NULL; UNIQUE; PRIMARY KEY e DEFAULT), ovviamente esistono anche i **VINCOLI DI DOMINIO**, che specificano i valori dei campi di tipo corretto.
- **Vincoli di Integrità INTERRELAZIONALI:** coinvolgono più tabelle ovvero vincoli di integrità referenziale (FOREIGN KEY)

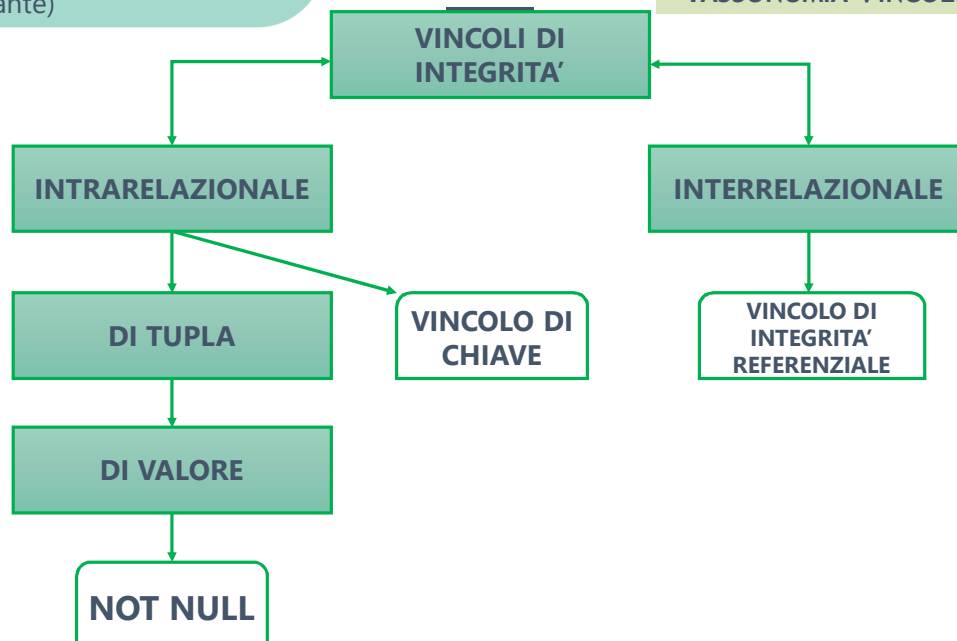
Quando vedi la tua verità fiorire sulle labbra del tuo nemico, devi gioire, perché questo è il segno della vittoria. (Giorgio Almirante)

BASI DI DATI

3

Concetti base – SQL

TASSONOMIA VINCOLI DI INTEGRITÀ



Sii sempre, in ogni circostanza e di fronte a tutti, un uomo libero e pur di esserlo sii pronto a pagare qualsiasi prezzo. (Sandro Pertini)

BASI DI DATI

4

Concetti base – SQL

VINCOLI PREDEFINITI SU ATTRIBUTI

- **[NOT NULL | NULL]**: In assenza del vincolo **NOT NULL**, l'attributo può contenere valori **NULL**;
- **[DEFAULT valore]**: Usato per impostare un valore di default compatibile con il suo dominio, utile quando in un inserimento il valore dell'attributo non viene specificato, in assenza si assume come valore di default **NULL**;
- **[UNIQUE | PRIMARY KEY]**: **UNIQUE** rappresenta un attributo che non può contenere valori duplicati (una chiave); **PRIMARY KEY**, indica la chiave primaria che oltre a non ammettere valori duplicati non può contenere valori **NULL**;
- **[REFERENCES Nome_Tabella [Nome_Attributo]]**: Consente di definire un vincolo di chiave esterna su *Nome_Attributo* verso la chiave primaria di *Nome_Tabella*

BASI DI DATI

5

Concetti base – SQL

VINCOLI PREDEFINITI SU TABELLA

- **[PRIMARY KEY (nome_attributo1, nome_attributo2...nome_attributoN)]**: Consente di definire una chiave primaria per la tabella utilizzando un insieme di attributi. E' fondamentale definire una e una sola PRIMARY KEY per ogni relazione
- **[UNIQUE (nome_attributo1, nome_attributo2...nome_attributoN)]**: Consente di definire una (Super)chiave candidata per la tabella utilizzando un insieme di attributi. Viene fatta eccezione per il valore NULL, che puo comparire su righe diverse senza violare il vincolo
- **[FOREIGN KEY (nome_a1, nome_a2...nome_aN) REFERENCES Nome_Tabella [nome_a1, nome_a2...nome_aN]]**: Consente di definire vincoli di chiave esterna su più attributi

BASI DI DATI

6

Concetti base – SQL

VINCOLI SU PIÙ ATTRIBUTI

- Si osservi che definizioni di una tabella ad esempio con le seguenti due modalità non producono la stessa cosa.

CREATE TABLE caso1 (

...

Nome CHAR(20) NOT NULL,

Cognome CHAR(20) NOT NULL,

UNIQUE (Cognome, Nome),); *(crea una chiave su nome e cognome ma con possibili duplicati di nomi o cognomi)*

CREATE TABLE caso2 (

...

Nome CHAR(20) NOT NULL UNIQUE,

Cognome CHAR(20) NOT NULL UNIQUE, ...); *crea una chiave per nome e una chiave per cognome, non sono possibili duplicati sui nomi o sui cognomi)*

BASI DI DATI

7

Concetti base – SQL

VINCOLI INTRARELAZIONALI

- **NOT NULL**
- **UNIQUE:** definisce (Super)chiavi, chiavi candidate
- **PRIMARY KEY:** chiave primaria (una sola, implica NOT NULL; DB2 non rispetta lo standard)
- **CHECK:** controlli di dominio, tra attributi di una tupla, ecc.
- ❖ *I Vincoli UNIQUE e PRIMARY KEY possono essere espressi in due forme*
 - Caso1) nella definizione di un attributo se forma da solo una chiave
 - Caso2) come elemento separato ad esempio
 - Caso1) **Matricola CHAR(6) PRIMARY KEY**
 - Caso2) **Matricola CHAR(6),**
 - ...,**
 - PRIMARY KEY (Matricola)**

BASI DI DATI

8

Concetti base – SQL

VINCOLI INTERRELAZIONALI

- **CHECK:** anche su più attributi di tabelle differenti (**warning**)
- **REFERENCES e FOREIGN KEY:** permettono di definire vincoli di integrità referenziale
- Sono possibili due sintassi per singoli attributi e su più attributi
- ❖ *E' possibile definire politiche di reazione alla violazione*

BASI DI DATI

9

Concetti base – SQL

CREATE TABLE: UN ESEMPIO

➤ Synopsis

```
CREATE TABLE Nome Tabella (
    NomeAttributo1 tipoDati(Dominio) [valore di default] [vincoli],
    { NomeAttributo1 tipoDati(Dominio) [valore di default] [vincoli] }
    AltriVincoli (es. di Tabella) );
```

➤ Esempio

```
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) );
```

BASI DI DATI

10

Concetti base – SQL

CREATE TABLE: UN ESEMPIO

➤ Synopsis

```
CREATE TABLE Nome Tabella (
    NomeAttributo1 tipoDati(Dominio) [valore di default] [vincoli],
    { NomeAttributo1 tipoDati(Dominio) [valore di default] [vincoli] }
    AltriVincoli (es. di Tabella) );
```

➤ Esempio

```
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) );
```

BASI DI DATI

11

Infrazioni

Concetti base – SQL

FOREIGN KEY

<u>Codice</u>	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

Tabella
Referenziate,
Slave, Interna
Secondaria

Vigili

<u>Matricola</u>	Cognome	Nome
3987	Rossi	Luca
3295	Neri	Piero
9345	Neri	Mario
7543	Mori	Gino

Tabella
Referenziata,
Master, Esterna,
Principale

BASI DI DATI

12

Infrazioni

Concetti base – SQL

FOREIGN KEY

<u>Codice</u>	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

Tabella
Referenziate,
Slave, Interna
Secondaria

Auto

<u>Prov</u>	<u>Numero</u>	Cognome	Nome
MI	39548K	Rossi	Mario
TO	E39548	Rossi	Mario
PR	839548	Neri	Luca

Tabella
Referenziata,
Master, Esterna,
Principale

BASI DI DATI

13

Concetti base – SQL

CREATE TABLE: UN ESEMPIO

➤ Synopsis

```
CREATE TABLE Nome Tabella (
    NomeAttributo1 tipoDati(Dominio) [valore di default] [vincoli],
    { NomeAttributo1 tipoDati(Dominio) [valore di default] [vincoli] }
    AltriVincoli (es. di Tabella) );
```

➤ 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));
```

BASI DI DATI

14

Concetti base – SQL

CREATE INDEX

➤ Gli indici rappresentano uno strumento prezioso per velocizzare l'accesso ai dati. Essi sono rilevanti dal punto di vista delle prestazioni, in passato rappresentavano l'unico modo per definire chiavi.

❖ **CREATE [unique] INDEX** nome_ind on NomeTable (ListaAttributi)

❖ **DROP INDEX** nome_ind

- ✓ **unique** indica che nella tabella non sono possibili righe che assumono lo stesso valore su ListaAttributi (che quindi è una chiave eventualmente non minimale)
- ✓ L'**ordine** in cui sono dati gli attributi è importante.

Esempio: create unique index ind on Anag (Cognome, Nome)

BASI DI DATI

15

Concetti base – SQL

COMANDI FONDAMENTALI

TABLE ↔ Tabella ↔ Relazione
 ROW ↔ Riga ↔ Tupla
 COLUMN ↔ Colonna ↔ Attributo

➤ **SQL come DDL ha tre comandi fondamentali:**

Create

Database, Schema, Table, Domain, Constraint, ...

Alter

Database, Schema, Table, Domain, Constraint, ...

Drop

Database, Schema, Table, Domain, Constraint, ...

BASI DI DATI

16

Concetti base – SQL

ALTER DATABASE

➤ **ALTER DATABASE**

ALTER DATABASE — change a database

➤ **Synopsis**

❖ **ALTER DATABASE** name [[WITH] option [...]]

where option can be:

ALLOW_CONNECTIONS allowconn

CONNECTION LIMIT connlimit

IS_TEMPLATE istemplate

❖ **ALTER DATABASE** name **RENAME TO** new_name

❖ **ALTER DATABASE** name **OWNER TO** { new_owner | **CURRENT_USER** | **SESSION_USER** }

❖ **ALTER DATABASE** name **SET TABLESPACE** new_tablespace

ALTER DATABASE name **SET** configuration_parameter { **TO** | **=** } { value | **DEFAULT** }

ALTER DATABASE name **SET** configuration_parameter **FROM CURRENT**

ALTER DATABASE name **RESET** configuration_parameter

ALTER DATABASE name **RESET ALL**

➤ **Esempio**

Per disabilitare le scansioni dell'indice per impostazione predefinita database test:

ALTER DATABASE test **SET** enable_indexscan **TO off**;second_column integer);

BASI DI DATI

17

Concetti base – SQL

ALTER DATABASE

- **ALTER DATABASE** modifica gli attributi di un database. Solo il proprietario del database o un superutente può modificare queste impostazioni.
- ❖ **Il primo comando** modifica alcune impostazioni per database.
- ❖ **La seconda forma** cambia il nome del database. Il database corrente non può essere rinominato.
- ❖ **Il terzo modulo** cambia il proprietario del database. Per modificare il proprietario, è necessario possedere il database ed essere anche un membro diretto o indiretto del nuovo ruolo proprietario e è necessario disporre del privilegio CREATEDB. (
- ❖ **La quarta forma** cambia lo spazio delle tabelle predefinito del database. È necessario disporre del privilegio di creazione per il nuovo tablespace. Questo comando sposta fisicamente tutte le tabelle o gli indici nel vecchio tablespace predefinito del database nel nuovo tablespace.

BASI DI DATI

18

Concetti base – SQL

DROP DATABASE

- **DROP DATABASE**
- DROP DATABASE — remove a database**
- **Synopsis**
- DROP DATABASE [IF EXISTS] name***
- ❖ DROP DATABASE rilascia un database. Rimuove le voci di catalogo per il database ed elimina la directory contenente i dati.
- ❖ Può essere eseguito solo dal proprietario del database.
- ❖ Non può essere eseguito mentre si è connessi al database di destinazione.
- ❖ **DROP DATABASE** non può essere annullato. **Usalo con cura!**

BASI DI DATI

19

Concetti base – SQL

ALTER DOMAIN

➤ ALTER DOMAIN

ALTER DOMAIN — change the definition of a domain

➤ Synopsis

- ❖ **ALTER DOMAIN** name { SET DEFAULT expression | DROP DEFAULT }
- ❖ **ALTER DOMAIN** name { SET | DROP } NOT NULL
- ❖ **ALTER DOMAIN** name ADD domain_constraint [NOT VALID]
- ❖ **ALTER DOMAIN** name DROP CONSTRAINT [IF EXISTS] constraint_name [RESTRICT | CASCADE]
- ❖ **ALTER DOMAIN** name RENAME CONSTRAINT constraint_name TO new_constraint_name
- ❖ **ALTER DOMAIN** name VALIDATE CONSTRAINT constraint_name
- ❖ **ALTER DOMAIN** name OWNER TO { new_owner | CURRENT_USER | SESSION_USER }
- ❖ **ALTER DOMAIN** name RENAME TO new_name
- ❖ **ALTER DOMAIN** name SET SCHEMA new_schema *Esempio*

BASI DI DATI

20

Concetti base – SQL

ALTER DOMAIN: ESEMPI

➤ Esempi

- Per aggiungere un vincolo NOT NULL a un dominio:
ALTER DOMAIN zipcode SET NOT NULL;
- Per rimuovere un vincolo NOT NULL da un dominio:
ALTER DOMAIN zipcode DROP NOT NULL;
- Per aggiungere un vincolo di controllo a un dominio:
ALTER DOMAIN zipcode ADD CONSTRAINT zipchk CHECK (char_length(VALUE) = 5);
- Per rimuovere un vincolo di controllo da un dominio:
ALTER DOMAIN zipcode DROP CONSTRAINT zipchk;
- Per rinominare un vincolo di controllo su un dominio:
ALTER DOMAIN zipcode RENAME CONSTRAINT zipchk TO zip_check;
- Per spostare il dominio in uno schema diverso:
ALTER DOMAIN zipcode SET SCHEMA customers;
- ❖ Si osservi che la versione modificata di un dominio deve essere soddisfatta dai dati già presenti. Diversamente la modifica è rifiutata

BASI DI DATI

21

Concetti base – SQL

ALTER DOMAIN: UN ESEMPIO

➤ ALTER DOMAIN

DROP DOMAIN — change the definition of a domain

➤ Esempio

Create Domain ETAIMPIEGATI As Integer -- dominio elementare

Default 31

Check (Value >= 18 and Value <= 80));

Create Table Impiegato (

Id Int,

Nome Varchar(20)

Età ETAIMPIEGATI);

- ❖ Se eseguiamo il comando **ALTER Domain ETAIMPIEGATI DROP Default**, Se il valore dell'età nelle tuple è stato inserito direttamente il comando va a buon fine, diversamente se deriva dal valore di default del nuovo dominio il comando fallisce in quanto la definizione della nuova versione della tabella non è più soddisfatta dai dati presenti.

BASI DI DATI

22

Concetti base – SQL

DROP DOMAIN

➤ DROP DOMAIN

DROP DOMAIN — remove a domain

➤ Synopsis

DROP DOMAIN [IF EXISTS] name [, ...] [CASCADE | RESTRICT]

- ❖ L'opzione **Restrict – usata di default** – specifica che il comando non deve essere eseguito in presenza di oggetti nella base dati che usano il dominio.

➤ Esempio

Create Domain ETAIMPIEGATI As Integer -- dominio elementare

Default 31

Check (Value >= 18 and Value <= 80));

Create Table Impiegato (

Id Int,

Nome Varchar(20)

Età ETAIMPIEGATI);

- ❖ Se eseguiamo il comando **DROP Domain ETAIMPIEGATI**, il dominio non viene rimosso perché è utilizzato nella definizione di Tabella Impiegato (Restrict).

BASI DI DATI

23

➤ DROP DOMAIN

Concetti base – SQL

DROP DOMAIN: UN ESEMPIO

DROP DOMAIN — remove a domain

➤ Esempio

**Create Domain ETAIMPIEGATI As Integer -- dominio elementare
Default 31**

Check (Value >=18 and Value <=80));

Create Table Impiegato (

Id Int,

Nome Varchar(20)

Età ETAIMPIEGATI);

- ❖ Se eseguiamo il comando **DROP Domain ETAIMPIEGATI CASCADE**, il dominio viene rimosso ed il dominio del campo Età diviene Integer ovvero il dominio elementare con cui è stato definito il dominio rimosso ETAIMPIEGATO.

BASI DI DATI

24

➤ ALTER TABLE

Concetti base – SQL

ALTER TABLE

ALTER TABLE — change the definition of a table

➤ Synopsis

ALTER ALTER TABLE [IF EXISTS] [ONLY] name [*] action [, ...]

Laddove *action* è una delle seguenti azioni (in elenco solo quelle usuali):

❖ **ADD [COLUMN] [IF NOT EXISTS] column_name data_type [COLLATE collation] [column_constraint [...]]**

✓ aggiunge una nuova colonna alla tabella, utilizzando la stessa sintassi di CREATE TABLE

❖ **DROP [COLUMN] [IF EXISTS] column_name [RESTRICT | CASCADE]**

✓ elimina una colonna da una tabella. Indici e vincoli di tabella che coinvolgono la colonna sono cancellati

❖ **ALTER [COLUMN] column_name [SET DATA] TYPE data_type [COLLATE collation] [USING expression]**

✓ cambia il tipo di una colonna di una tabella, effettuando se possibile un Cast implicito diversamente utilizzare USING con in expression le modalità per la modifica dei valori.

❖ **ALTER [COLUMN] column_name SET DEFAULT expression**

❖ **ALTER [COLUMN] column_name DROP DEFAULT**

✓ Questi comandi impostano o rimuovono il valore predefinito per una colonna. I valori predefiniti si applicano solo dopo con i comandi INSERT o UPDATE, quindi non causa la modifica delle righe esistenti.

✓ **ALTER [COLUMN] column_name { SET | DROP } NOT NULL**

✓ Queste forme cambiano se una colonna è contrassegnata per consentire o per rifiutare valori NULL si può usare o per rifiutare valori NULL. _Usare SET NOT NULL solo quando la colonna non contiene valori NULL.

BASI DI DATI

25

Concetti base – SQL

ALTER TABLE

➤ ALTER TABLE

ALTER TABLE — change the definition of a table

- **Synopsis**, oltre a **ALTER TABLE [IF EXISTS] [ONLY] name [*] action [, ...]** vi sono altre forme:

ALTER TABLE [IF EXISTS] [ONLY] name [*] RENAME [COLUMN] column_name TO new_column_name

ALTER TABLE [IF EXISTS] [ONLY] name [*] RENAME CONSTRAINT constraint_name TO new_constraint_name

ALTER TABLE [IF EXISTS] name RENAME TO new_name

ALTER TABLE [IF EXISTS] name SET SCHEMA new_schema

BASI DI DATI

26

Concetti base – SQL

ALTER TABLE: ESEMPI

➤ Esempi

- Per aggiungere una colonna di tipo varchar a una tabella:

ALTER TABLE distributors ADD COLUMN address varchar(30);

- Per eliminare una colonna da una tabella:

ALTER TABLE distributors DROP COLUMN address RESTRICT;

- Per modificare i tipi di due colonne esistenti in un'unica operazione:

ALTER TABLE distributors ALTER COLUMN address TYPE varchar(80), ALTER COLUMN name TYPE varchar(100);

- Per modificare una colonna intera contenente timestamp Unix in timestamp con fuso orario tramite una clausola USING:

ALTER TABLE foo ALTER COLUMN foo_timestamp SET DATA TYPE timestamp with time zone USING timestamp with time zone 'epoch' + foo_timestamp * interval '1 second';

- Per rinominare una colonna esistente:

ALTER TABLE distributors RENAME COLUMN address TO city;

BASI DI DATI

27

Concetti base – SQL

ALTER TABLE: ESEMPI

- Esempi
- Per rinominare una tabella esistente:
ALTER TABLE distributors RENAME TO suppliers;
- Per rinominare un vincolo esistente:
ALTER TABLE distributors RENAME CONSTRAINT zipchk TO zip_check;
- Per aggiungere un vincolo NOT NULL a una colonna:
ALTER TABLE distributors ALTER COLUMN street SET NOT NULL;
- Per rimuovere un vincolo NOT NULL da una colonna:
ALTER TABLE distributors ALTER COLUMN street DROP NOT NULL;
- Per aggiungere un vincolo di controllo a una tabella e a tutti i suoi figli:
ALTER TABLE distributors ADD CONSTRAINT zipchk CHECK (char_length(zipcode) = 5);
- Per aggiungere un vincolo di controllo solo a una tabella e non ai suoi figli:
ALTER TABLE distributors ADD CONSTRAINT zipchk CHECK (char_length(zipcode) = 5) NO INHERIT; (Il vincolo check non verrà ereditato nemmeno dai figli futuri.)

BASI DI DATI

28

Concetti base – SQL

ALTER TABLE: ESEMPI

- Esempi
- Per rimuovere un vincolo di controllo da una tabella e da tutti i suoi figli:
ALTER TABLE distributors DROP CONSTRAINT zipchk;
- Per rimuovere un vincolo di controllo da una sola tabella:
ALTER TABLE ONLY distributors DROP CONSTRAINT zipchk; (Il vincolo check rimane in vigore per tutte le tabelle figlie.)
- Per aggiungere un vincolo di chiave esterna a una tabella:
ALTER TABLE distributors ADD CONSTRAINT distfk FOREIGN KEY (address) REFERENCES addresses (address);
- Per aggiungere un vincolo di chiave esterna a una tabella con il minimo impatto su altro lavoro:
ALTER TABLE distributors ADD CONSTRAINT distfk FOREIGN KEY (address) REFERENCES addresses (address) NOT VALID;
ALTER TABLE distributors VALIDATE CONSTRAINT distfk;
- Per spostare una tabella in un altro tablespace:
ALTER TABLE distributors SET TABLESPACE fasttablespace;

BASI DI DATI

29

Concetti base – SQL

ALTER TABLE: ESEMPI

➤ Esempi

- Per aggiungere un vincolo univoco (a più colonne) a una tabella:

```
ALTER TABLE distributors ADD CONSTRAINT dist_id_zipcode_key UNIQUE (dist_id, zipcode);
```

- Per aggiungere un vincolo di chiave primaria denominato automaticamente a una tabella, notando che una tabella può avere solo una chiave primaria:

```
ALTER TABLE distributors ADD PRIMARY KEY (dist_id);
```

- Per spostare una tabella in uno schema diverso:

```
ALTER TABLE myschema.distributors SET SCHEMA yourschema;
```

- Per ricreare un vincolo di chiave primaria, senza bloccare gli aggiornamenti durante la ricostruzione dell'indice:

```
CREATE UNIQUE INDEX CONCURRENTLY dist_id_temp_idx ON distributors (dist_id);
```

```
ALTER TABLE distributors DROP CONSTRAINT distributors_pkey,  
ADD CONSTRAINT distributors_pkey PRIMARY KEY USING INDEX  
dist_id_temp_idx;
```

BASI DI DATI

30

Concetti base – SQL

DROP TABLE

➤ DROP TABLE

DROP TABLE — remove a table

➤ Synopsis

```
DROP TABLE [ IF EXISTS ] name [, ...] [ CASCADE | RESTRICT ]
```

- ❖ DROP TABLE rimuove le tabelle dal database. Solo il proprietario della tabella, e/o dello schema e/o il superutente possono eliminare una tabella. Per svuotare una tabella di righe senza distruggerla, usa DELETE o TRUNCATE.
- ❖ DROP TABLE rimuove sempre tutte le righe (struttura e contenuti), gli indici, le regole, i trigger e i vincoli esistenti per la tabella di destinazione.
- ❖ Per eliminare una tabella a cui fa riferimento una vista o un vincolo di chiave esterna di un'altra tabella, è necessario specificare CASCADE. (Ovviamente CASCADE rimuoverà del tutto una vista dipendente, ma nel caso della chiave esterna rimuoverà solo il vincolo di chiave esterna e non l'altra tabella.)

➤ Esempio

DROP Table films, distributors;

BASI DI DATI

31

Concetti base – SQL

CREATE/ALTER/DROP TABLE: ESEMPI

➤ Esempi

➤ Per Creare una tabella Genitori con due attributi Figlio e Genitore:

CREATE TABLE Genitori (
Figlio varchar(20),
Genitore varchar(20);

➤ Per aggiungere una Primary Key composta da Figlio e Genitore

ALTER TABLE Genitori ADD Primart Key (Figlio, Genitore);

➤ Per aggiungere un vincolo di Foreign Key all'attributo Figlio della relazione Genitori. Tale attributo referencia l'attributo nome che è chiave primaria della Tabella Persone

ALTER TABLE Genitori ADD Foreign Key (Figlio) References Persone(Nome)

➤ Per rimuovere la tabella Persone

DROP TABLE Persone CASCADE;

BASI DI DATI

32

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)