


# Laboratorio di Basi di Dati 2018-2019

## I esercitazione: SQL DDL e creazione di schemi

Il testo proposto per l'esercitazione (come di tutte quelle che saranno proposte nel corso) è lungo. Se non riuscite a completare tutti i punti, potete completare le esercitazioni al di fuori le due ore del laboratorio, per esercitarvi ulteriormente sull'argomento.

### Documentazione di riferimento

- Introduzione a PostgreSQL e laboratorio:  
<https://2018.aulaweb.unige.it/mod/resource/view.php?id=54242>
- Introduzione a SQL DDL:  
<https://2018.aulaweb.unige.it/mod/resource/view.php?id=54239>
- Manuale online PostgreSQL: <http://www.postgresql.org/docs/manuals/>
- Lanciare PgAdmin III 
- Collegarsi al server [webdev.disi.unige.it](http://webdev.disi.unige.it) con le credenziali assegnate  
Username: **bd\_gruppoXX**  
Password: **bd\_gruppoXX**  
I numeri dei gruppi li trovate qui:  
<https://2018.aulaweb.unige.it/mod/resource/view.php?id=66026>  
(nota XX vale 01...09 per i gruppi da 1 a 9)
- Nella schermata principale, la parte sinistra (object browser) visualizza i database, gli schemi e gli oggetti dello schema (tabelle, vincoli, ...).
- Espandere con + il proprio database.  
Il database si chiama **DB\_BDuserXX**
- Sotto Schemas ci sarà il solo schema public.  
  
In generale è buona norma utilizzare uno schema diverso per ogni dominio/applicazione, quindi anche se non è strettamente necessario e potreste lavorare sotto public il consiglio è di creare un nuovo schema per l'esercitazione.
  - Per creare un nuovo schema si utilizza il comando **CREATE SCHEMA "nome\_schema";** (es. **create schema "labo1";**)
- **FARE ATTENZIONE:** lo schema di default è public (quindi crea lì le tabelle). Per spostarvi in un altro schema si usa il comando  
**set search\_path to "nome\_schema";** (es. **set search\_path to "labo1";**)
- Ogni volta che aprite una nuova finestra dell'editor comandi SQL ricordatevi di settare correttamente lo schema utilizzando questo comando.
- Aprire l'editor comandi SQL (icona SQL dentro a lente di ingrandimento)
- **Nota 1:** Salvare su file le istruzioni eseguite nell'esercizio 1. Sarà quanto dovrete consegnare nella consegna del laboratorio.
- **Nota 2:** pgAdmin è bandwidth friendly, in generale, per poter visualizzare l'effetto di una modifica nell'Object Browser potrebbe essere necessario un refresh.

### ESERCIZIO 0

- Aprite (da editor SQL pgAdmin, con Open e non cliccandoci sopra perché all'estensione .sql è associato un programma diverso da pgAdmin) ed eseguite lo script SQL **creazioneVideoteca.sql** che trovate su AulaWeb.
- Esaminate il contenuto dello script e, attraverso l'Object Browser esaminate gli effetti della sua esecuzione. In particolare, il contenuto delle tabelle si può esaminare posizionandosi sulla tabella (espandendo lo schema e Tables con +) e selezionando View data dal menu che si apre cliccando il tasto destro del mouse.
- Prestate attenzione ai comandi insert into che vengono utilizzati per inserire le tuple nelle tabelle.

## ESERCIZIO 1

### 0. Creare uno schema “corsi” e posizionarsi in tale schema eseguendo set search\_path to “corsi”.

1. Creare la relazione che memorizza i dati relativi ai professori dell'Università di Genova con schema

**Professori(Id, Cognome, Nome, Stipendio, InCongedo)**

dove

- Id rappresenta l'identificativo univoco, numerico di 5 cifre, di ogni professore, ed è chiave primaria della relazione;
- Cognome e Nome sono il cognome e il nome di ogni professore, devono essere obbligatoriamente presenti e costituiscono una chiave alternativa (unique) per la relazione;
- Stipendio rappresenta lo stipendio annuale in Euro di ogni professore (utilizzare 8 cifre totali di cui due decimali) e ha valore di default di 15000 Euro;
- InCongedo, a valori Booleani, specifica se un professore è in congedo, e ha valore predefinito FALSE.

Utilizzare i tipi di dato numeric e char varying, ragionando sulle dimensioni più opportune per la definizione di ciascun campo. Verificare che la tabella sia effettivamente presente e verificare che abbia lo schema che si voleva ottenere.

Dopo aver creato la tabella Professori:

- (a) Inserire alcune tuple (almeno 5) nella tabella, facendo attenzione a rispettare i vincoli.

Il comando è `INSERT INTO Professori VALUES`, ad es.

`INSERT INTO Professori VALUES (54661, 'guerrini', 'giovanna', 123456.78, FALSE);`

Trovate ulteriori esempi nello script creazioneVideoteca.sql

- (b) Provare ad inserire 3 tuple all'interno della relazione Professori, cercando di violare i vincoli di chiave primaria, unicità e di not null e facendo attenzione ai messaggi di errore di PostgreSQL. **Gli inserimenti che non vengono eseguiti perché violano vincoli e i corrispondenti messaggi di errori vanno inseriti nel file SQL da consegnare come commenti (iniziare la riga con due trattini –).**

2. Creare la relazione che mantiene le informazioni sugli insegnamenti (o corsi) dei vari corsi di laurea dell'Università di Genova con schema

**Corsi(Id, CorsoDiLaurea, Nome, Professore, Attivato)**

- Id rappresenta un identificativo (univoco) alfanumerico di 10 caratteri di ogni corso ed è chiave primaria della relazione;
- CorsoDiLaurea rappresenta il corso di laurea (es. Informatica, Smid, Matematica) in cui il corso è offerto;
- Nome è il nome del corso (es. Basi di Dati);
- Professore è l'identificativo del professore titolare del corso ed è chiave esterna verso la relazione Professori;
- Attivato è un attributo Booleano, con valore predefinito FALSE, che indica se il corso è attivato nell'anno accademico corrente oppure no.

Nome e corso di laurea sono sempre indicati, mentre Professore e Attivazione potrebbero essere nulli.

L'integrità referenziale rispetto alla chiave esterna Professore deve essere gestita in modo tale che:

- se viene modificato l'identificativo di professore per una tupla di Professori, la modifica sia propagata alla tabella Corsi;
- non sia permesso di cancellare un Professore se esistono corsi associati a quel professore.

- (a) Inserire 5 tuple nella tabella Corsi, relative a corsi di Informatica, Elettronica, Programmazione, Fisica, Matematica. Tenere presenti le tuple di Professori inserite al punto precedente e provare a inserire tuple che causano violazioni del vincolo di integrità referenziale (assegnando un identificatore di professore che non esiste). Inserire anche corsi non attivati e senza professore.
- (b) Effettuare la modifica e la cancellazione di una tupla dalla tabella Professori, così da verificare il comportamento a seguito di violazioni del vincolo di integrità referenziale.

### 3. Creare la relazione che mantiene i dati degli studenti dell'Università di Genova con schema

#### Studenti(Matricola, Cognome, Nome, CorsoDiLaurea, Iscrizione, Relatore)

- Matricola rappresenta l'identificativo, numerico progressivo, di ogni studente ed è chiave primaria della relazione; si vuole che i valori della colonna corrispondente vengano generati automaticamente dal DBMS<sup>1</sup>;
- Cognome e Nome rappresentano il cognome e il nome di ogni studente;
- CorsoDiLaurea rappresenta il corso di laurea (es. informatica, matematica) a cui lo studente è iscritto;
- Iscrizione rappresenta l'anno accademico (es. 2006/2007) della prima iscrizione al corso di laurea;
- Relatore è il professore (si ipotizzi che sia uno solo) che segue il lavoro di tesi dello studente e che viene assegnato allo studente durante il corso dei suoi studi. L'attributo può assumere valore nullo per gli studenti che non abbiano ancora assegnato un relatore.

Ragionare sui vincoli da utilizzare per mantenere la consistenza dei dati memorizzati e sui tipi di dati più adatti per implementare gli attributi. Notare che Relatore è chiave esterna per la relazione: una modifica ai dati del relatore deve aver effetto anche sui dati degli studenti che segue, mentre non deve essere possibile cancellare i dati di un professore se esistono studenti di cui è relatore.

Inserire alcune tuple nella tabella Studenti verificando che i vincoli definiti siano mantenuti e facendo attenzione ai comportamenti impostati per le violazioni dell'integrità referenziale.

Provare a modificare le tabelle definite, in particolare provare a

- (a) Modificare la tabella Corsi aggiungendo un attributo<sup>2</sup> opzionale MutuaDa alfanumerico di 10 caratteri che costituisce una chiave esterna su Corsi e rappresenta l'identificatore del corso da cui il corso è eventualmente mutuato. Specificare il comportamento che si ritiene più opportuno in caso di violazioni dell'integrità referenziale. Inserire una nuova tupla corrispondente al corso di Basi di dati per Smid che mutua da Basi di dati per Informatica.
- (b) Modificare la tabella Professori modificando la colonna Stipendio<sup>3</sup> in modo che possa contenere dati con 9 cifre totali (di cui 2 decimali).
- (c) Modificare nuovamente la tabella Professori, modificando la colonna Stipendio, in modo che possa contenere dati con 5 cifre non decimali e due decimali. Osservare il messaggio restituito da PostgreSQL.
- (d) Dopo aver inserito in Corsi le tuple (68, 'Informatica', 'tcs', NULL, NULL) e (69, 'Informatica', 'tcs', NULL, NULL), modificare le proprietà della tabella Corsi in modo che all'interno della tabella non siano ammessi corsi con lo stesso nome e stesso corso di laurea<sup>4</sup>. Provare a salvare le modifiche apportate e annotare i messaggi di errore di PostgreSQL. Dopodiché modificare le tuple della tabella Professori che violano il vincolo di unicità appena inserito così che lo rispettino e salvare le modifiche apportate alla struttura della tabella.
- (e) Modificare la tabella Professori, cancellando la colonna InCongedo<sup>5</sup>.
- (f) Modificare le tabelle definite aggiungendo i vincoli CHECK su colonna e su riga che si ritengono necessari.

---

<sup>1</sup> Si utilizzi a questo scopo il tipo SERIAL.

<sup>2</sup> Il comando è ALTER TABLE Corsi ADD COLUMN MutuaDa tipo\_colonna.

<sup>3</sup> Il comando è ALTER TABLE Professori ALTER COLUMN Stipendio TYPE nuovo\_tipo\_colonna.

<sup>4</sup> Il comando è ALTER TABLE Professori ADD CONSTRAINT nome\_vincolo def\_vincolo.

<sup>5</sup> Il comando è ALTER TABLE Professori DROP COLUMN nome\_di\_colonna.