

Laboratorio di Basi di Dati

Esercitazione PostgreSQL

Dopo aver lanciato il client grafico pgAdmin III di PostgreSQL svolgere le operazioni descritte nel seguito, tenendo presenti i suggerimenti forniti unitamente al testo dell'esercitazione.

Le operazioni si possono svolgere utilizzando sia le funzionalità dell'interfaccia grafica che l'interfaccia per la definizione dei comandi di SQL di pgAdmin III. Si consiglia di provare ad utilizzare entrambe le modalità. Per le operazioni effettuate mediante interfaccia grafica si consiglia di visionare comunque il codice SQL generato.

Si raccomanda di salvare su un file di testo i comandi che verranno definiti ed eseguiti per lo svolgimento di questa esercitazione.

Svolgere i seguenti passi:

1. Utilizzando le funzionalità fornite dall'interfaccia grafica, creare un nuovo Database con nome "esercizio1". Osservare nel riquadro SQL (in basso a destra) i comandi generati dall'interfaccia.
2. All'interno dello Schema public, creare la relazione che memorizza i dati relativi ai professori dell'Università di Milano, con schema:

Professore(id, **Cognome**, **Nome**), 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 e devono essere obbligatoriamente presenti.

Utilizzare i tipi di dato **numeric** e **char varying**, ragionando sulle dimensioni più opportune per la definizione di ciascun campo.

3. Dopo aver creato la tabella **Professore**:
 - (a) Modificare la tabella **Professore**, aggiungendo alla tabella le colonne **Stipendio**, che rappresenta lo stipendio annuale in Euro di ogni professore (utilizzare 7 cifre totali di cui due decimali) che non può ovviamente assumere valori negativi e che ha valore predefinito di 35000 Euro, e la colonna **InCongedo**, a valori booleani, che specifica se un professore è in congedo, con valore predefinito false.
 - (b) Inserire almeno quattro tuple nella tabella.
 - (c) Modificare la tabella **Professore** modificando la colonna **Stipendio** in modo che possa contenere dati con 8 cifre totali (di cui 2 decimali).
 - (d) Modificare nuovamente la tabella **Professore**, modificando la colonna **Stipendio**, in modo che possa contenere dati con 3 cifre non decimali e due decimali. Osservare il messaggio restituito da PostgreSQL. Il comando va a buon fine?

- (e) Inserire altre tuple nella tabella **Professore**, inserendo dei valori per la colonna **InCongedo** e facendo attenzione a rispettare i vincoli di chiave primaria e di not null. Inserire due professori che abbiano lo stesso nome e cognome.
 - (f) Modificare le proprietà della tabella **Professore** in modo che all'interno della tabella non siano ammessi professori con lo stesso nome e cognome. Provare a salvare le modifiche apportate e annotare i messaggi di errore di PostgreSQL. Dopodiché modificare le tuple della tabella **Professore** che violano il vincolo di unicità appena inserito così che lo rispettino e salvare le modifiche apportate alla struttura della tabella.
 - (g) Provare ad inserire delle tuple all'interno della relazione **Professore**, cercando di violare i vincoli di chiave primaria, di unicità e di not null.
 - (h) Modificare la tabella **Professore**, cancellando la colonna **InCongedo**.
 - (i) Modificare la tabella **Professore**, cancellando il vincolo su **Stipendio**.
4. Creare la seguente tabella, tenendo presenti la descrizione degli attributi ed i vincoli specificati nel seguito:
- (a) si consideri la relazione che mantiene le informazioni sugli insegnamenti (corsi) dei vari corsi di laurea dell'Università di Milano, con schema: **Corso**(id, **CorsoDiLaurea**, **Denominazione**, **Professore**, **Attivato**)
 - **id** rappresenta l'identificativo alfanumerico di 4 caratteri, univoco, di ogni corso, ed è chiave primaria della relazione;
 - **CorsoDiLaurea** rappresenta il corso di laurea (es. ingegneria, informatica) in cui il corso è tenuto;
 - **Denominazione** è il nome del corso;
 - **Docente** è l'identificativo del professore titolare del corso, ed è chiave esterna verso la relazione **Professore**;
 - **Attivato** è un attributo booleano, con valore predefinito false, che indica se il corso è attivato nell'anno accademico corrente oppure no.

La denominazione del corso e il corso di laurea sono sempre indicate e la coppia **CorsoDiLaurea** e **Denominazione** è unica all'interno della relazione.

Se un corso è attivato, allora deve essere indicato il professore titolare del corso.

[**SUGGERIMENTO:** utilizzare un vincolo cosiddetto 'di riga' che specifica che per ogni tupla **attivato** vale 'F' oppure **professore** non è nullo: `check (Attivato='F' or professore is not null).`]

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 **Professore**, la modifica sia propagata alla tabella **Corsi**;
 - non sia permesso di cancellare un **Professore** se esistono corsi associati a quel professore.
- (b) Inserire 5 tuple nella tabella **Corsi**, relative a corsi di Informatica, Elettronica, Programmazione, Fisica, Matematica. Tenere presenti le tuple di **Professore** 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.
- (c) Effettuare la modifica e la cancellazione di una tupla dalla tabella **Professore**, così' da verificare il comportamento a seguito di violazioni del vincolo di integrità' referenziale.
5. Si consideri la relazione che mantiene i dati degli studenti dell'Università' di Milano con schema:

Studente(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;
- **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. 2014/2015) della prima iscrizione al corso di laurea;
- **Relatore** è il professore che segue il lavoro di tesi dello studente e che viene assegnato allo studente durante il corso dei suoi studi.

Creare la tabella per questa relazione, ragionando sui vincoli da utilizzare per mantenere la consistenza dei dati memorizzati e sui tipi di dati più adatti per implementare gli attributi. Si noti che **Relatore** è chiave esterna per la relazione; la semantica dell'attributo richiede che una modifica ai dati del relatore abbia effetto anche sui dati degli studenti che segue durante la tesi di laurea, mentre che non sia possibile cancellare i dati di un professore se esistono studenti di cui è stato relatore.

Inserire almeno 7 tuple nella tabella **Studenti** verificando attentamente che i vincoli definiti siano effettivamente mantenuti e facendo attenzione ai comportamenti impostati per le violazioni dell'integrità' referenziale. Nel caso di propagazione per vedere le modifiche al contenuto di una tabella è necessario chiudere e riaprire la tabella.

6. Scrivere in SQL ed eseguire dall'interfaccia di interrogazione di pgAdmin (osservare i risultati restituiti da PostgreSQL), i comandi che permettano di ottenere:
- (a) tutti i dati di tutti gli studenti dell'Università di Milano;
 - (b) i corsi attivati durante l'anno accademico in corso per tutti i corsi di laurea dell'Università di Milano;
 - (c) cognome, nome e anno di iscrizione degli studenti di informatica dell'Università di Milano, in ordine alfabetico;
 - (d) l'elenco degli studenti dei corsi di laurea di informatica che non hanno assegnato alcun relatore;
 - (e) i nomi, in ordine alfabetico inverso e senza duplicati, dei corsi tenuti in tutti i corsi di laurea dell'Università di Milano;
 - (f) i dati dei corsi di cui si conosce il professore titolare;
 - (g) gli identificativi e i nomi dei corsi di 'Informatica' di cui si conosce il professore titolare attivati nell'anno;
 - (h) i nominativi dei professori che guadagnano uno stipendio compreso fra 40000 e 60000 euro l'anno;
 - (i) i nominativi dei professori, con stipendio annuo compreso fra 40000 e 60000 euro, unitamente al loro stipendio, ordinando i risultati secondo valori decrescenti di stipendio annuo;
 - (j) la matricola, il cognome e il nome degli studenti dell'Università di Milano iscritti ai corsi di laurea di informatica o di matematica;
 - (k) la matricola e il corso di laurea degli studenti iscritti a fisica nell'anno accademico 2003/2004 o che hanno il cognome che comincia per 'P';
 - (l) cognome e nome degli studenti di informatica dell'Università di Milano iscritti dopo l'anno accademico 2002/2003, il cui nome sia compreso, in senso alfabetico, fra 'Beatrice' e 'Giovanni';