

Esercitazione 01

Si vuole progettare il sistema informativo per la gestione dei musei del Veneto. Una possibile traccia dello schema relazionale è dato dal seguente sezione.

Schema base di dati

Le entità di seguito specificate conterranno le informazioni relative ai musei e alle opere ritenute significative.

```
Museo(nome: stringa(30 car.),  
      città: stringa(20 car.),  
      indirizzo: ...,  
      numeroTelefono: ...,  
      giornoChiusura: ...,  
      prezzo: ...,  
      )
```

```
Opera(nome: stringa(30 car.),  
      cognomeAutore: stringa(20 car.),  
      nomeAutore: stringa(20 car.),  
      museo: ...,  
      città: ...,  
      epoca: ...,  
      anno: ...  
      )
```

```
Mostra(titolo: stringa(30 car.),  
       inizio: DATE,  
       fine: DATE,  
       museo: ...,  
       città: ...,  
       prezzo: ...,  
       )
```

```
Orario(progressivo: intero,  
       museo: ...,  
       città: ...,  
       giorno: ..., //proporre un dominio!  
       orarioApertura: TIME WITH TIME ZONE,  
       orarioChiusura: TIME WITH TIME ZONE  
       )
```

- Gli attributi chiave primaria sono sottolineati.
- Gli attributi *museo* e *città* nelle entità **Opera**, **Mostra** e **Orario** sono chiave esportata. La chiave esterna è nella tabella **Museo**.
- **Attributi diversi da null:** *Museo.giornoChiusura*, *Museo.prezzo*, *Mostra.inizio*, *Mostra.fine*, *Orario.museo*, *Orario.città*, *Orario.giorno*.
- **Valori di default:** *Museo.città* ha come valore di default 'Verona'. *Museo.nome* ha come valore di default 'MuseoVeronese'. *Museo.prezzo* ha come valore di default 10. *Orario.orarioApertura* ha come valore di default '09:00 CET'. *Orario.orarioChiusura* ha come valore di default '19:00 CET'.
- **Politiche di reazione da specificare per FOREIGN KEY:**
 - Per **Opera** usare **CASCADE** in caso di modifica degli attributi e **SET NULL** in caso di cancellazione della riga.
 - Per **Mostra** usare **SET DEFAULT** (usando i dati di un museo esistente nella tabella museo) in caso di modifica degli attributi e in caso di cancellazione della riga.
 - Per **Orario** usare **CASCADE** in caso di modifica dell'attributo e anche in caso di cancellazione della riga.
- Per l'inserimento di date e orario, fare riferimento al capitolo 8.5.1 del manuale di PostgreSQL 9.6.

- Tutte le risposte ai seguenti esercizi dovranno essere scritte in un file eseguibile nel server PostgreSQL.
- Usare i commenti SQL per documentare. Esempio: *-- Questo è un commento!*
- Per ciascuna scelta di implementazione (tipo la scelta del dominio degli attributi), il file DEVE contenere un commento in italiano che motivi la scelta fatta.
- Il codice deve essere indentato e le parole chiave di SQL devono essere sempre in maiuscolo. Un esempio:

```
CREATE TABLE prova (  
    nome VARCHAR(80) NOT NULL,  
    cognome VARCHAR(80) NOT NULL,  
    UNIQUE(nome, cognome)  
);
```
- Il file dovrà contenere anche, come commento SQL all'inizio file, le seguenti informazioni:
 1. il vostro nominativo composto da cognome, nome e numero matricola.
 2. quale esercizio deve essere valutato dal tutor rispetto a una vostra domanda o dubbio circa la correttezza.
- Come prova di correttezza sintattica, eseguire più volte il file direttamente dal comando 'psql':

```
psql -U <userid> -f <nomeFile> <base di dati>
```
- Concordare con il tutor come inviare il file contenente il codice.
- Il tutor esegue il file prima di rispondere. Se si verifica un errore di esecuzione NON previsto, il tutor non risponde alla domanda e segnala lo studente al professore del corso.

Esercizio 1

Scrivere il codice PostgreSQL che generi tutte le tabelle. Per gli attributi di cui non è stato specificato il tipo, scegliere quello opportuno. Specificare tutti i vincoli possibili, sia intra- sia inter-relazionali.

Esercizio 2

Inserire nell'entità **Museo** le seguenti tuple:

```
(Arena, Verona, piazza Bra, 045 8003204, martedì, 20),  
(CastelVecchio, Verona, Corso Castelvecchio, 045 594734, lunedì, 15);
```

Esercizio 3

Popolare le tabelle **Opera** e **Mostra** con almeno altre tre tuple ciascuna.

Esercizio 4

Provare ad inserire nella relazione **Museo** tuple che violino i vincoli specificati.

Esercizio 5

Nell'entità **Museo**, aggiungere l'attributo *sitoInternet* e inserire gli opportuni valori.

Esercizio 6

Nell'entità **Mostra** modificare l'attributo *prezzo* in *prezzoIntero* ed aggiungere l'attributo *prezzoRidotto* con valore di default 5. Aggiungere il vincolo (di tabella o di attributo?) che garantisca che *Mostra.prezzoRidotto* sia minore di *Mostra.prezzoIntero*.

Esercizio 7

Nell'entità **Museo** aggiornare il *prezzo* aggiungendo 1 Euro alle tuple esistenti.

Esercizio 8

Nell'entità **Mostra** aggiornare il *prezzoRidotto* aumentandolo di 1 Euro per quelle mostre che hanno *prezzoIntero* inferiore a 15 Euro.