

## Esercitazione 09

---

### Interrogazioni in MongoDB

Utilizzando le slide presentate nell'ultima lezione di laboratorio, creare un account su Atlas e connettersi ad un cluster MongoDB, quindi creare un database per la gestione dei musei del Veneto, equivalente a quello creato in PostgreSQL durante la prima esercitazione di laboratorio.

Nello specifico, una volta creato il database, connettersi ad esso tramite Compass ed eseguire i seguenti esercizi.

### Esercizio 1

Creare una collezione Museo ed inserire alcuni documenti con la seguente struttura(usare le funzioni insertOne() or insertMany()) eventualmente facendo copia ed incolla da un file di testo → **Attenzione** il caricamento tramite file JSON richiede di specificare un ObjectId per ciascun documento, al contrario l'inserimento tramite insertOne() o insertMany() crea tale identificatore in automatico e lo ritorna come risultato dell'inserimento).

Per gli attributi di tipo data usare il tipo ISODate("") in modo da poter eseguire delle ricerche.

```
Museo {
  nome: '...',
  citta: '...',
  indirizzo: '...',
  numeroTelefono: '...',
  giornoChiusura: '...',
  prezzo: ...,
  opere: [
    { nome: '...',
      cognomeAutore: '...',
      nomeAutore: '...',
      epoca: '...',
      anno: '...',
    },
    ...
  ],
  mostre: [
    { titolo: '...',
      inizio: '...',
      fine: '...',
      prezzo: ...,
    },
    ...
  ],
}
```

```

    orari: [
      { giorno: '...',
        orarioApertura: ...,
        orarioChiusura: ...,
      },
      ...
    ]
  }

```

## Esercizio 2

Creare una collezione MuseoRef equivalente alla precedente, in cui però le proprietà opere e mostre usano riferimenti tramite ObjectId('...') a documenti presenti in altre due collezioni Opera e Mostre con la seguente struttura.

```

Opera {
  nome: '...',
  cognomeAutore: '...',
  nomeAutore: '...',
  epoca: '...',
  anno: '...',
}
Mostra {
  titolo: '...',
  inizio: '...',
  fine: '...',
  prezzo: ....,
}

```

## Esercizio 3

Cercare tutti i musei che si trovano a Verona, riportando solo il nome ed il numero di telefono.

Attenzione: la funzione .project() può essere utilizzata solo all'interno di applicazioni (es. python) ed in presenza di un cursore. Al contrario nell'interazione tramite shell (Compass oppure mongosh) non è possibile usare la funzione .project(), ma è possibile specificare la proiezione come secondo argomento della funzione find(), come segue:

```
db.<collection>({ item: 'value'}, { item1: 1, item2: 1 })
```

in questo modo solo gli attributi item1 e item2 saranno visualizzati.

### Esercizio 4

Cercare tutti i musei che sono aperti il martedì oppure il giovedì.

L'interrogazione su una proprietà innestata avviene nel seguente modo:

'orari.giorno': 'Martedì'

Ricordarsi di usare le virgolette singole per indicare il nome della proprietà innestata.

### Esercizio 5

Aggiornare i prezzi dei musei di Verona aggiungendo un prezzo ridotto pari a 5 euro.

La funzione update può essere utilizzata sia per aggiornare un valore, che aggiornare la struttura di un documento aggiungendo una proprietà mancante.

Verificare l'esito eseguendo una interrogazione che mostri solo nome del museo, città, prezzo e prezzo ridotto.

### Esercizio 6

Selezionare tutti i musei che hanno una mostra attiva oggi, visualizzando il nome del museo ed il nome della mostra.

Usare gli operatori \$gte e \$lte con il tipo ISODate().

### Esercizio 7

Utilizzare l'operatore di lookup per eseguire un join tra ciascun documento Museo ed i corrispondenti documenti Mostra e Opera. Selezionare solo i musei di Verona.