

# Serie 1

Nome e cognome: \_\_\_\_\_ Classe: \_\_\_\_\_

## Esercizio 1

Creare un file che contiene le informazioni per connettersi alla banca dati, il file deve essere strutturato in modo tale da poter essere utilizzato nei diversi script che si andranno a creare nella serie.

Provare pure a svilupparlo sotto forma di classe.

## Esercizio 2

Creare un form nel quale posso inserire delle query SQL e visualizzare il risultato ritornato:

- Stato
- Errore

Interpretare in maniera corretta gli errori.

## Esercizio 3

Creare un form con i seguenti campi:

- Nome (testo, max 50 caratteri) (\*)
- Cognome (testo, max 50 caratteri) (\*)
- Via (testo, max 100 caratteri)
- CAP (numero, 4 cifre) (\*)
- Città (testo, 35 caratteri) (\*)
- Nazione (testo, 50 caratteri) (\*)
- Telefono (testo o numero)
- Email (testo, 50 caratteri) (\*)
- Osservazioni (testo a lunghezza indefinita)
- Sesso (enum m o f) (\*)
- Hobby (elenco a scelta multipla)

Utilizzare gli elementi più appropriati per visualizzare il form, i campi con (\*) sono obbligatori. Gestire il form (errore) ed **inserire** i dati in una tabella in MySQL. Scrivere il codice in modo procedurale ed ad oggetti (2 versioni).

## Esercizio 4

Creare lo schema UML (a scelta) delle classi per la gestione della connessione con la banca dati. Al momento attuale i requisiti minimi sono:

- gestire la connessione con la banca dati di tipo MySQL
- gestire eventuali errori di connessione
- gestire l'inserimento di dati nel db
- gestire gli errori d'inserimento
- creazione/modifica/eliminazione di tabelle
- creazione/modifica/eliminazione di campi

Sono da prevedere in futuro degli ampliamenti.

## Esercizio 5

Scrivere le classi dell'esercizio 4.

## Esercizio 6

Espandere l'esercizio 5 creando i metodi per inserire i dati nel db senza dover scrivere la query. Creare un metodo **prepareInsert** il quale prepara i dati per l'inserimento nel database.

## Esercizio 7

Creare la seguente tabella da vostro codice:

```
CREATE TABLE `test`.`contatti` (  
  `id` INT NOT NULL AUTO_INCREMENT PRIMARY KEY ,  
  `nome` VARCHAR( 50 ) NOT NULL ,  
  `cognome` VARCHAR( 50 ) NOT NULL ,  
  `via` VARCHAR( 50 ) NULL ,  
  `cap` INT NOT NULL ,  
  `citta` VARCHAR( 50 ) NOT NULL ,  
  `nazione` VARCHAR( 50 ) NOT NULL ,  
  `email` VARCHAR( 50 ) NULL ,  
  `telefono` VARCHAR( 50 ) NULL ,  
  `cellulare` VARCHAR( 50 ) NULL  
) ENGINE = InnoDB
```

Creare un form di inserimento (utilizzando quanto fatto nella serie precedente) gestendo tutte le eccezioni possibili come ad esempio i campi obbligatori, ...

Eseguire la query per visualizzarne tutto il contenuto (stampare il contenuto in una forma intelleggibile, ad esempio in una tabella).

## Esercizio 8

Creare un FORM di ricerca per i dati presenti nell'esercizio 7, eseguire la QUERY di ricerca e stampare i risultati sotto forma di tabella.

## Esercizio 9

Espandere l'esercizio 8 permettendo di eseguire delle ricerche “avanzate”, cioè dove è possibile parametrizzare meglio la tipologia di ricerca attraverso ad esempio gli operatori logici AND e OR, il carattere jolly %, ...

## Esercizio 10

Gestire le diverse eccezioni che possono intervenire nell'esercizio 3, come ad esempio la mancanza di record trovati, la mancanza di parametri di ricerca, ...

## Esercizio 11

Data una QUERY di selezione tipo

```
SELECT * FROM contatti;
```

stampare i risultati in una pagina implementando una “paginazione alternativa”, cioè integrando le seguenti funzionalità:

- numero definito di record per pagina (da impostare nel file di configurazione sotto forma di costante);
- pulsante o link per spostarsi al primo record
- pulsante o link per spostarsi all'ultimo record
- pulsanti o link per spostarsi in avanti e indietro

Indicare pure il numero totale di record e la posizione attuale. Implementare pure il sistema di paginazione offerto dal motore di ricerca Google.



## **Esercizio 12**

Integrare l'esercizio 9 e 11, FORM di ricerca e stampa dei risultati con pulsanti di navigazione.

## **Esercizio 13**

Se non ancora fatto integrare le diverse funzionalità viste negli esercizi precedenti sotto forma di classe/i.

## **Esercizio 14**

Integrare quanto fatto nell'esercizio 13 con quanto fatto nella serie precedente per le operazioni di INSERT.