

Basi di Dati

SVILUPPO DI UN APPLICAZIONE WEB PER POPOLARE UNA BASE DI DATI

La finalità di questa esercitazione è quella di creare, date delle specifiche progettuali, appositi script di creazione e popolamento di una base di dati.

1. Istruzioni preliminari per la creazione, il popolamento e la creazione della base di dati

1. Installare il software EASYPHP 5.3.8.1 scaricabile dal sito Web <http://www.easyphp.org/download.php>
2. Lanciare EasyPHP dalla barra di Avvio di Windows
3. Cliccare sulla freccia in basso a destra sulla barra delle applicazioni, poi con il tasto destro sull'icona di EasyPHP
4. Aprire l'interfaccia *Administration* e poi cliccare su "Open" alla voce "MySQL Administration: PhpMyAdmin", sotto la scritta "Modules", per interfacciarsi con il DBMS MySQL
5. Per lanciare uno script SQL dall'interfaccia Web MySQL selezionare dal pannello *Importa* selezionare il file e cliccare su *Esegui*
6. Per rilanciare più volte lo script di creazione/popolamento ricordarsi di cancellare eventuali istanze del database create in precedenza dal pannello *Database*
7. Per scrivere/eseguire una query attraverso l'interfaccia Web di MYSQL selezionare il pannello *SQL*
8. Una volta verificata la correttezza delle interrogazioni salvarsi l'istruzione eseguita in un opportuno file di testo *query.txt* per poterla reinserire successivamente nel codice dell'applicazione Web (Nota: è possibile aggiungere l'istruzione tra i *Preferiti* attraverso il pannello *SQL*)

2. Istruzioni preliminari per lo sviluppo dell'applicazione Web in PHP

1. Le pagine HTML e PHP sono file di testo scritti con un qualsiasi editor (es., Notepad, Word, Wordpad).
2. I file contenenti i comandi PHP vanno salvati con estensione *.php* e copiati nella cartella *data/localweb* che si trova all'interno della cartella principale d'installazione del programma EasyPHP (ad es: *C:\Program Files (x86)\EasyPHP-x.x.x.x*).
3. Per navigare le pagine HTML/PHP, cliccare sulla freccia in basso a destra sulla barra delle applicazioni, poi selezionare *Local Web* dal menù accessibile cliccando sul tasto destro dell'icona EasyPHP.
4. I comandi PHP vanno inseriti dal corpo della pagina HTML e racchiusi all'interno dei tag di apertura `<?php` e chiusura `?>`
5. Ogni istruzione PHP termina con il simbolo ";
6. Qualora vengano eseguite istruzioni accesso alla base di dati, verificare di aver creato e popolato correttamente il DB in MySQL prima di accedere alle pagine.

3. Esercizi

Lanciare gli script *creaDB.sql* e *popolaDB.sql*, creati durante l'esercitazione precedente, relativi alla creazione e al popolamento della base di dati.

- Scrivere l'istruzione SQL per realizzare la seguente operazione.

Transazione di inserimento dati nella base di dati

Definire una nuova attività con la relativa programmazione.

- Creare un'applicazione web in PHP che consenta l'inserimento di nuove attività e della relativa programmazione. L'applicazione deve permettere di effettuare la procedura d'inserimento corrispondente all'istruzione SQL scritta per la Parte I.

Il form di inserimento dati deve

1. richiedere tutti i dati necessari all'inserimento di una nuova attività nella base di dati e quelli relativi alla sua programmazione.
2. permettere la selezione del codice fiscale dell'istruttore deve avvenire tramite un menù a tendina generato dal contenuto della tabella del database.

La pagina di risultato deve

1. valutare il corretto inserimento dei dati da parte dell'utente verificando:
 - a. se sono stati indicati tutti i campi
 - b. **tramite la creazione di funzioni** se si verificano inconsistenze tra i valori indicati come orario d'inizio e di fine attività (ad esempio, ora d'inizio 18:30 e ora di fine 15:00¹), o nei tipi di dato relativi al livello o alla sala (ad esempio, alfanumerici anziché numerici). In presenza di errori la pagina deve notificare il relativo errore.
 - c. se la sala selezionata, nel giorno e nell'ora selezionate è disponibile. In presenza di errori la pagina deve notificare il relativo errore.
 - d. se l'istruttore, nel giorno e nell'ora selezionate non è occupato in un'altra attività. In presenza di errori la pagina deve notificare il relativo errore.
2. Valutare se l'inserimento termina correttamente (ad esempio il codice dell'attività inserita esiste già). La pagina deve visualizzare un messaggio di corretto inserimento, altrimenti deve notificare l'errore.

Le Figure n.2 e n.3 mostrano un esempio di form e del risultato.

¹In PHP, le date o le ore possono essere **verificate** (controllo del tipo) e **confrontate** (controllo di inconsistenze) usando la funzione `strtotime($stringa)`, che permette di convertire un orario fornito in formato stringa in uno in formato time e quindi confrontabile. Se la stringa inserita NON è in un formato corretto la funzione ritorna FALSE, altrimenti la stringa convertita in formato time.

Se la data è corretta i risultati (time) possono essere confrontati utilizzando gli operatori di confronto (> < =).

Programmazione x +

← → ↻ 127.0.0.1:8888/istruttori/inserisci_attivita/form.php

Inserimento nuova programmazione

Dati della nuova attività da inserire

Codice:

Tipo:

Nome:

Livello:

Programmazione della nuova attività

Giorno:

Ora inizio:

Ora fine:

Sala:

Codice istruttore:

Figura 2. Esempio di un form

Programmazione x +

← → ↻ 127.0.0.1:8888/istruttori/inserisci_attivita/inserisci.php?codice=FITNPIL&ti

L'attività FITNPIL è stata inserita nel database ed è programmata per il giorno Lunedì.

Figura 3. Esempio del risultato