

Basi di Dati

Quaderno 4: Progetto di una base di dati e realizzazione di un'applicazione web

1. Creazione della base di dati

Realizzare la base di dati per la gestione di un campionato ciclistico considerando lo schema logico riportato di seguito.

CICLISTA (CodC, Nome, Cognome, Nazionalita, CodS, AnnoNascita)

SQUADRA (CodS, NomeS, AnnoFondazione, SedeLegale*)

TAPPA (Edizione, CodT, CittaPartenza, CittaArrivo, Lunghezza, Dislivello, GradoDifficolta)

CLASSIFICA_INDIVIDUALE (CodC, CodT, Edizione, Posizione)

Nota: i campi sottolineati identificano la chiave primaria di ciascuna relazione mentre gli asterischi indicano i campi opzionali. Alcuni dettagli sui campi delle varie tabelle sono riportati di seguito:

- CodT è un intero positivo incrementale (1, 2, etc.).
- Edizione è un intero positivo incrementale (1, 2, etc.).
- Lunghezza è espressa in metri.
- Posizione contiene la posizione di arrivo del ciclista in tale tappa (1, 2, etc.).
- GradoDifficolta è un intero positivo compreso tra 1 e 10.
- Dislivello è espresso in metri.
- AnnoFondazione e AnnoNascita sono valori interi compresi tra 1900 e 2000.
- Tutti i campi testuali hanno una lunghezza massima di 50 caratteri.

Sono richieste le seguenti attività:

- Creare uno script SQL (creaDB.sql) con:
 - Le istruzioni per la creazione della base di dati corrispondente allo schema logico indicato e la specifica degli opportuni vincoli d'integrità.
 - Le istruzioni per il popolamento della base di dati creata al punto precedente (almeno un record per ogni tabella). I dati possono essere inseriti a piacimento dallo studente.
- Scrivere le istruzioni SQL per realizzare le seguenti operazioni:
 - *Interrogazione della base dati realizzata:*
Fissato il codice di un ciclista e il codice di una tappa (a vostra scelta), visualizzare l'elenco delle posizioni ottenute dal ciclista nelle varie edizioni. Per ogni ciclista visualizzare nome, cognome, nome della squadra di appartenenza e gli attributi identificativi della tappa scelta. Ordinare i risultati per ordine crescente di edizione.
 - *Popolamento della base di dati:*
Inserire nella base di dati i dati relativi ad un nuovo ciclista e ad una posizione nella classifica personale. Gestire opportunamente gli inserimenti tramite l'uso delle transazioni.

2. Funzioni utili

Interazione con il DBMS (maggiori dettagli disponibili sulle [slides del corso](#)):

Instaurare la connessione:

[`mysqli_connect\(\$host, \$username, \$password, \$dbname\)`](#)

Verifica dell'avvenuta connessione: [`mysqli_connect_errno\(\)`](#)

Esecuzione di una richiesta in modalità immediata: [`mysqli_query\(\$conn, \$query\)`](#)

Verifica numero di tuple nel risultato: [`mysqli_num_rows\(\$result\)`](#)

Recupero dei risultati: [`mysqli_fetch_array\(\$result\)`](#) per ottenere tutti i risultati bisogna iterare usando una struttura del tipo:

```
while ($row = mysqli_fetch_array($result)) {  
    #utilizzo della variabile $row  
}
```

Verifica tipo di dato:

Verifica se la variabile inserita è stata settata: [`isset\(\$var\)`](#)

Verificare che il parametro inserito non sia vuoto con una verifica di questo tipo:

```
if ($_REQUEST["parametro"] == "") {  
    echo '<p> ERRORE campo vuoto. </p>';  
    exit;  
}
```

Verifica se la variabile inserita è di tipo numerico: [`is_numeric\(\$var\)`](#)

Verifica se la variabile inserita è un intero: [`is_integer\(\$var\)`](#)

3. Progettazione e sviluppo dell'applicazione web

Progettare un'applicazione web in PHP per la consultazione online della base di dati realizzata in precedenza. L'applicazione deve permettere di effettuare due attività corrispondenti alle istruzioni SQL scritte per la Sezione 1.

Nota bene: Tutti i parametri delle eventuali richieste GET devono coincidere con i nomi degli attributi dello schema logico, come nel seguente esempio (attenzione alle maiuscole):

`inserimento.php?CodC=1&Nome=Marco&Cognome=Pantani&AnnoDiNascita=1970&Nazionalita=Italiana&CodS=5`

3.1 Interrogazione della base dati realizzata

Dopo aver selezionato il codice di un ciclista (da un menu a tendina popolato con i valori recuperati dalla base di dati) e il codice di una tappa (campo di testo libero), visualizzare l'elenco delle posizioni ottenute dal ciclista nelle varie edizioni. Per ogni ciclista visualizzare nome, cognome e nome della squadra di appartenenza. Deve essere generato un messaggio di errore specifico in caso di dati mancanti (almeno uno dei parametri vuoto) o non digitati correttamente (verificare che il codice ciclista e il codice tappa siano valori interi).

Di seguito un esempio del form da compilare per l'interrogazione:

Posizione Ciclista in Tappa

Codice Ciclista:

2 - Chris Froome

Codice Tappa:

2

Cerca

Figura 1. Esempio form per interrogazione della base dati

3.2 Inserimento nella base dati realizzata

Si devono creare due form: uno per l'inserimento di un nuovo ciclista e uno per l'inserimento di un posizionamento di un ciclista in una tappa (preesistente):

- Il primo form deve richiedere tutti i dati necessari per l'inserimento di un nuovo ciclista. Prima dell'inserimento è necessario:
 - Verificare che i tipi di dato inseriti siano corretti (verificare che il codice ciclista, codice squadra e l'anno di nascita siano valori interi positivi).
 - Verificare che la squadra per la quale si richiede l'inserimento esista già nella base dati.
 - Accertarsi che non ci siano già ciclisti con lo stesso codice.
- Il secondo form richiede tutti i dati necessari per l'inserimento di una entry nella tabella `CLASSIFICA_INDIVIDUALE`. In particolare, si richiede che:
 - I campi identificativi della tappa e del ciclista siano popolati usando i dati già presenti nella base dati e il posizionamento del ciclista sia inserito attraverso un campo di testo libero.
 - Verificare che i tipi di dato inseriti siano corretti (codice ciclista, codice tappa, edizione e posizione devono essere valori interi positivi).
 - Verificare che i dati siano consistenti (il ciclista con quel codice esista già nella base dati così come il codice tappa e l'edizione).
 - Verificare che non esista già una posizione per il ciclista nella tappa selezionata.

In caso di errore deve essere visualizzato un errore con testo bianco su sfondo rosso, altrimenti il messaggio di avvenuto inserimento deve essere visualizzato con un testo bianco su sfondo verde. Di seguito un esempio dei due form e dei messaggi di errore e corretto inserimento. Per la modifica dello stile possono essere utilizzate delle direttive CSS da stampare direttamente dal codice PHP (https://www.w3schools.com/css/css_colors.asp, [slides del corso](#)).

Inserisci Ciclista

Codice Ciclista:	<input type="text" value="e.g. 5"/>
Nome Ciclista:	<input type="text" value="e.g. Mario"/>
Cognome Ciclista:	<input type="text" value="e.g. Rossi"/>
Nazionalità Ciclista:	<input type="text" value="e.g. Italiana"/>
Codice Squadra:	<input type="text" value="Seleziona una Squadra"/>
Anno di Nascita Ciclista:	<input type="text" value="e.g. 1980"/>
<input type="button" value="Inserisci"/>	

Figura 2. Esempio form per inserimento ciclista.

Inserisci posizione in classifica

Codice Ciclista:	<input type="text" value="Seleziona un ciclista"/>
Codice Tappa:	<input type="text" value="Seleziona una tappa"/>
Edizione:	<input type="text" value="Seleziona un'edizione"/>
Posizione Ciclista:	<input type="text" value="e.g. 2"/>
<input type="button" value="Inserisci"/>	

Figura 3. Esempio form per inserimento posizione in classifica personale.

Inserimento avvenuto!

Inserimento avvenuto con successo.

Figura 4. Esempio messaggio positivo.

Errore!

La posizione per questo ciclista in questa tappa esiste già nella base dati.

Figura 5. Esempio messaggio d'errore.