

Relazione per progetto esame POIS/PMO di Marco Marcheggiani

INDICE

SPECIFICA.....	PAGINA 2
STUDIO DEL PROBLEMA.....	PAGINA 3
SCHEMA DELLE CLASSI.....	PAGINA 5
DOCUMENTAZIONE SULL'UTILIZZO.....	PAGINA 6
USE CASES.....	PAGINA 7

SPECIFICA

Si intende realizzare un Form per il calcolo dei risultati di gare di tiro con l'arco.

Le gare in esame appartengono al campionato F.I.T.A.S.T. ma il Form rimarrà abbastanza libero da poter essere utilizzato anche da altre federazioni arcieristiche.

Il Form permetterà l'inserimento di dati quali:

- Nome e Cognome dell'arciere
- Compagnia di appartenenza
- Categoria di appartenenza
- Tipo di arco utilizzato
- Luogo della gara
- Anno di svolgimento della gara
- Punteggio per ogni singola freccia in gara (3 frecce a bersaglio per 20 bersagli)
- Tipo di bersaglio (suddiviso in speciale/libero e non)

Il Form, tramite apposito pulsante, calcolerà il totale del punteggio di gara, il totale degli spot realizzati dall'arciere (centri perfetti) e la media punti.

Il Form sarà in grado di collegarsi con un database locale dove l'utente potrà:

- Salvare i dati inseriti sul Form
- Caricare dei dati già salvati
- Cancellare dei dati già salvati

STUDIO DEL PROBLEMA

Il semplice inserimento dei dati nel form per il calcolo dei risultati non è questione problematica. Basterebbe un semplice ciclo sulle:

- caselle contenenti i punteggi delle frecce
 - caselle a spunta che indicano i bersagli speciali/liberi
- per avere i tre totali interessati.

Per le operazioni su database invece, non basterebbe il ciclo.

Ho scelto quindi di utilizzare delle liste dove appoggiare i dati per poter svolgere i calcoli ed i “passaggi” a funzioni per la manipolazione dei dati. Questi dati all’interno delle liste vanno poi manipolati in maniera accorta in quanto il ciclo sopra le caselle del form avviene in senso opposto (es: i risultati vengono memorizzati in ordine dalla terza freccia del bersaglio 20 fino alla prima freccia del bersaglio 1).

Uno dei punti macchinosi del calcolo del punteggio è il controllo sul valore inserito. Il regolamento delle gare indica che, i bersagli standard possono assumere solo alcuni punteggi (0-2-5-8), mentre i liberi e gli speciali hanno delle normative diverse (liberi qualsiasi punteggio max 24 a bersaglio, speciali 0-5-10-15).

Visti i continui cambi di regolamento a riguardo, assieme alla possibilità di usare il programma in altre federazioni dove i punteggi sono differenti, mi ha spinto a non mettere vincoli al valore dei punteggi da inserire nelle caselle (inserire 9-9-9 come punteggio non genererà errore anche se non è un punteggio corretto).

Il lavorare con un database crea qualche difficoltà nella stesura del codice dovuta a vari fattori:

- tipi di dati e loro conversione tra MySQL e c# (es: in MySQL non esiste bool)
- certezza di stessa scrittura tra form e database (problema case sensitive)
- gestione delle chiavi del database
- connessione al database nelle fasi di salvataggio, ricerca e cancellazione

Tipi di dati e loro conversione tra MySQL e c#

I tipi di dato bool di c# e boolean di MySQL non sono corrispondenti così come non lo sono i menù a tendina di tipo string di c# e i char set di MySQL.

Il tipo boolean di MySQL corrisponde infatti ad un intero ad una cifra in cui 0 = ‘false’ altrimenti vale ‘true’. Questa incompatibilità porta ad utilizzare i tipi int per le operazioni booleane sia su c# che su MySQL.

Per i menù a tendina del form non riconosciuti da MySQL è bastato il setting su database a tipo varchar. Questa modifica comporta che, eseguendo una query direttamente dall’editor del database, il valore dei menù potrebbe non essere tra

quelli consentiti (es: il tipo arco potrebbe essere 'pippo' invece di uno dei tre indicati nel form). Fortunatamente le query vengono svolte dal form e quindi i record possono assumere solo i valori stabiliti sul form stesso.

Anche il tipo DateTime di c# non è compatibile con il tipo DATE di MySql. Anche in questo caso sono ricorso all'utilizzo di interi per indicare l'anno della gara (una città organizza massimo una gara all'anno).

Certezza di stessa scrittura tra form e database (problema case sensitive)

Un utente potrebbe salvare i dati di gara dell'arciere 'Pippo' nella città di 'Urbino' nell'anno '2020'.

Cosa potrebbe succedere se il giorno dopo volesse caricare o cancellare i dati dell'arciere 'PIPP0' relativi alla gara di 'URBINO' del '2020'?

Ovviamente la query non troverebbe nessun record in quanto sia c# che MySql sono case sensitive.

Per garantire un corretto svolgimento delle operazioni durante il salvataggio, caricamento e cancellazione da database, i campi sono stati uniformati tramite ToUpper() in modo che non possano esserci errori.

Gestione delle chiavi del database

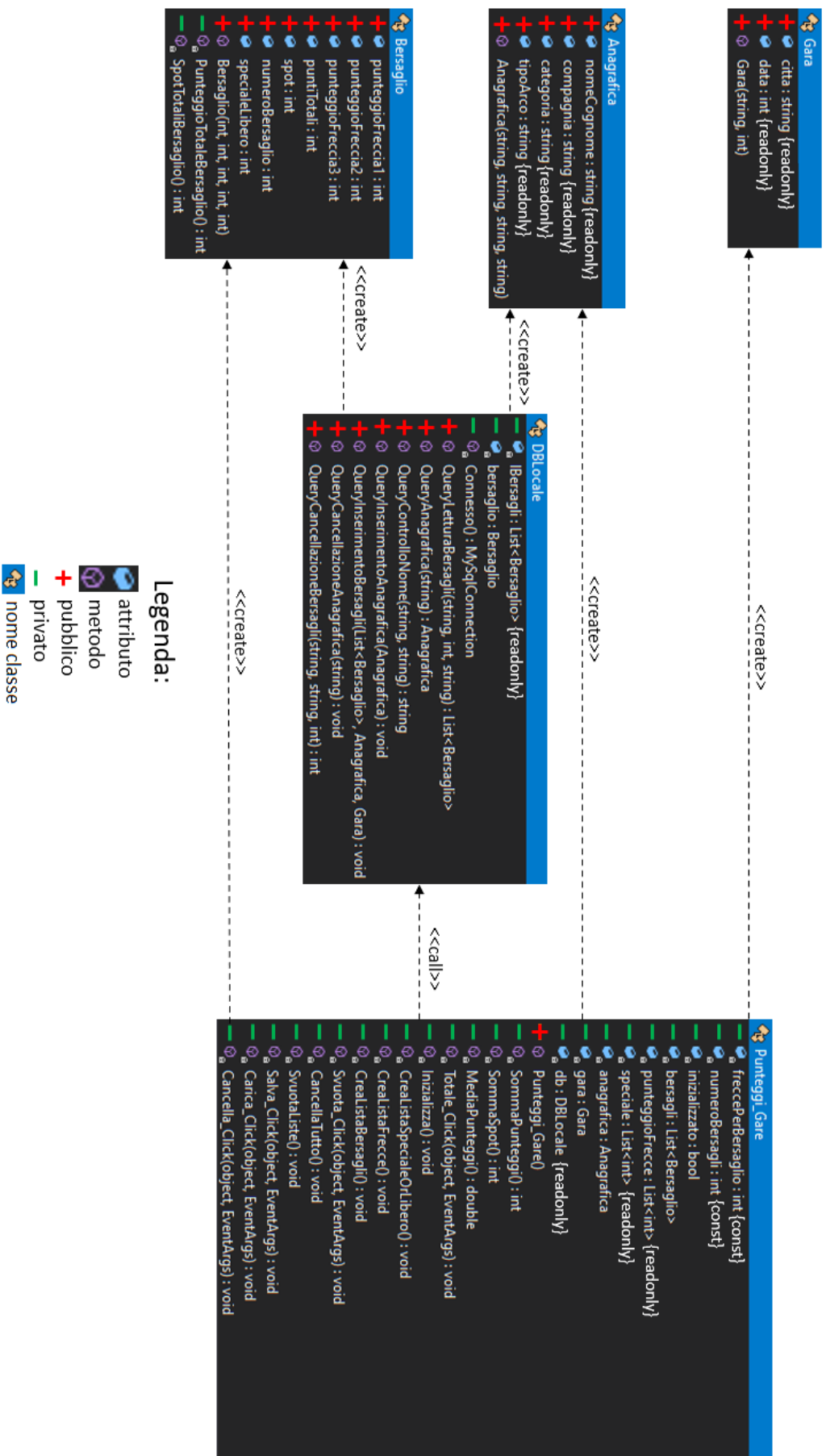
Una delle tabelle del database possedeva una chiave primaria combinata a tre campi. Provando una query direttamente sul database, il vincolo di chiave funzionava come previsto ma questo non si ripeteva, facendo eseguire la query dal form.

In base a questa costruzione, il programma leggeva come duplicati le gare svolte da uno stesso arciere, in luoghi differenti o in anni differenti. Dopo varie soluzioni tentate ho deciso di rimuovere il vincolo di chiave primaria dalla tabella.

Connessione al database nelle fasi di salvataggio, ricerca e cancellazione

Il programma per connettersi ad un database MySql deve prevedere i pacchetti per le operazioni con MySql. Non bastano i pacchetti base per la connessione a SQL e database associati.

SCHEMA DELLE CLASSI



DOCUMENTAZIONE SULL'UTILIZZO

Senza database abbinato il programma può solo calcolare i totali dei punteggi inseriti nel form.

Il database deve chiamarsi 'risultati' e deve essere accessibile tramite localhost senza password. In caso si voglia modificare questo requisito si devono modificare i dati della stringa `"server=localhost;user=root;password=;database=risultati"` nel metodo `Connesso()` della classe `DBLocale`.

Le query che comunicano e lavorano con il database utilizzano sintassi ad hoc per la soluzione riportata. Nel momento in cui vengano modificati i nomi dei campi delle tabelle del database, andranno modificate anche le query della classe `DBLocale`.

USE CASES

