Università degli Studi di Bologna

Facoltà di Ingegneria e Scienze Informatiche (A.A. 2014/2015)

*Studenti:*

*Fabio Pazzini - Matricola: 0000692345 - email:* *fabio.pazzini3@studio.unibo.it*

*Aldo Dushku - Matricola: 0000691687 - email: aldo.dushku@studio.unibo.it*

*Relazione per il C*orso di Basi di Dati

Progetto di una Base di Dati per la

gestione di un Campionato

Regionale di Karate

Indice

Analisi dei requisiti

Lo scopo di questo progetto è la realizzazione di una Base di Dati per coordinare e mantenere i dati relativi ad un Campionato Regionale di Karate, che si deve svolgere in una certa palestra. Il progetto considera le varie prove effettuate da determinati atleti, mostrando attenzione anche alla parte di giuria, a cui fanno parte arbitri, giudici ed un tavolo di giuria. La gara prevede l’esecuzione di un Kata su un Tatami (parte di pavimento destinata alla prova), allo svolgimento della quale viene dato un voto dall’arbitraggio. Ogni atleta appartiene, inoltre, ad una specifica categoria e proviene da una certa palestra, così come gli arbitri e i giudici.

Il programma prevede due tipi di utilizzazione: una in modalità “root”, che consente l’aggiunta, la modifica e la rimozione di un certo elemento, ed una in modalità “user” che permette solamente operazioni di ricerca.

Il progetto non rispecchia totalmente una situazione realistica in quanto manca la gestione dei turni, eliminatorie e finali, e premiazioni.

La realizzazione del Database è stata effettuata utilizzando il DBMS PHPMyAdmin, con il linguaggio MySQL, mentre la parte relativa alla comunicazione, alle query, e a qualsiasi altro aspetto di interazione con esso è stata eseguita scrivendo codice Java e utilizzando le librerie del JDBC. Le query, invece, sono scritte in SQL.

Requisiti in linguaggio naturale

“*L’associazione sportiva di Karate KarateKid richiede un sistema informatico per la gestione del Torneo Regionale di Karate. Il sistema deve gestire la manifestazione come segue: una gara si deve svolgere in una determinata palestra (1\*), e prevede di sostenere una prova che consiste nell’esecuzione di un Kata alla quale gli arbitri daranno una votazione. La prova verrà coordinata da un tavolo di giuria al quale fanno parte dei giudici che provengono, come atleti e arbitri, da una certa palestra (2\*). Tutte le categorie possono partecipare, e vogliamo che venga visualizzato il primo classificato in base ai voti (3\*). Gli atleti eseguiranno la prova su un tatami, che è la parte di pavimentazione destinata alla gara. Della manifestazione si vogliono mantenere la data e la palestra in cui si è svolta (4\*).*

*Si vuole mantenere lo storico riguardante l’appartenenza di un certo atleta ad una determinata categoria (5\*).*

*L’amministratore di sistema, identificato dalla password “root”, deve avere la possibilità di accedere al Database e di modificarlo aggiungendo, rimuovendo o modificando elementi a sua scelta; l’utente finale deve invece avere la possibilità di eseguire solo delle ricerche all’interno della Base di Dati.”*

*1\* : Una palestra può ospitare più gare, mentre una gara può essere ospitata da una ed una sola palestra.*

*2\* : Si deve evitare che un giudice o un arbitro partecipi alla gara.*

*3\* : Per ogni categoria si vuole visualizzare il primo classificato.*

*4\* : In una certa data, e nella stessa palestra, non si possono svolgere due gare (anche con nomi diversi).*

*5\* : Un atleta può passare da una categoria all’altra nel corso degli anni.*

Scelta della strategia di Progetto

Riscriviamo il testo del Progetto e sottolineiamo i concetti fondamentali, quelli che ci consentiranno di realizzare un primo schema scheletro del nostro elaborato.

“*L’associazione sportiva di Karate KarateKid richiede un sistema informatico per la gestione del Torneo Regionale di Karate. Il sistema deve gestire la manifestazione come segue: una gara si deve svolgere in una determinata palestra (1\*), e prevede di sostenere una prova che consiste nell’esecuzione di un Kata alla quale gli arbitri daranno una votazione. La prova verrà coordinata da un tavolo di giuria al quale fanno parte dei giudici che provengono, come atleti e arbitri, da una certa palestra (2\*). Tutte le categorie possono partecipare, e vogliamo che venga visualizzato il primo classificato in base ai voti (3\*). Gli atleti eseguiranno la prova su un tatami, che è la parte di pavimentazione destinata alla gara. Della manifestazione si vogliono mantenere la data e la palestra in cui si è svolta (4\*).*

Sviluppo delle entità

Entità PARTECIPANTE:
Dato che arbitri, atleti e giudici di partecipano alla gara, può essere conveniente introdurre una generalizzazione a cui daremo il nome di PARTECIPANTE. Un generico partecipante sarà identificato dal proprio Codice Fiscale, e di esso verranno mantenute informazioni anagrafiche quali nome, cognome, luogo di nascita, data di nascita. Questi dati verranno ereditati dalle singole entità ATLETA, ARBITRO e GIUDICE. Dell’atleta si vogliono anche mantenere informazioni legate al sesso e al Dan (grado della cintura nera).

Entità PALESTRA:

Una palestra è identificata dal proprio nome e dalla palestra in cui si trova, e da una palestra provengono vari partecipanti, indipendentemente che essi siano atleti, arbitri o giudici. Di una palestra si mantiene anche l’informazione riguardante l’indirizzo presso cui si trova.

Entità PROVA:

L’entità PROVA è una delle più importanti entità dello schema, in quanto rappresenta la singola prova eseguita da un determinato atleta. La prova, come specificato nel testo del Progetto, prevede l’esecuzione di un Kata e l’assegnazione di un voto da parte degli arbitri. Di conseguenza, imposteremo le cardinalità in modo tale da avere come identificatore dell’entità la composizione del nome della prova, del Codice Fiscale dell’atleta e dal nome del Kata eseguito. Di una prova si vuole anche mantenere il Codice Fiscale dell’arbitro che ha giudicato e del voto che ha dato. Inoltre, per mantenere lo storico delle prove e delle categorie abbiamo incluso anche la chiave importata dell’entità CATEGORIA.

Dal momento che un prova si esegue su un Tatami, mentre su un Tatami si possono svolgere N prove, importiamo l’identificatore di Tatami che è il suo codice univoco.

Entità CATEGORIA, TAVOLO DI GIURIA, GARA, KATA:

Queste, essendo enti minori, le tratteremo tutte insieme. KATA e CATEGORIE sono entità che contengono un solo campo e le abbiamo inserite per facilitare il controllo dell’inserimento di una prova e di un atleta (un atleta appartiene ad una certa categoria e deve eseguire un Kata, esistenti nel Database).

La GARA è l’entità che rappresenta il contesto dell’applicativo ed è identificata dalla composizione tra la data e la palestra in cui si tiene, mentre TAVOLO DI GIURIA è l’oggetto che rappresenta il tavolo che presiede e coordina ogni tatami, composto da 5 giudici. Ogni tavolo è identificato da un proprio codice; all’interno del nostro Progetto non ha particolare importanza, ma ci è sembrato più opportuno conservare la giuria poiché in un Torneo reale essa occupa il ruolo di verificare la presenza di irregolarità causate da errori umani.

Lo schema concettuale

Schema di navigazione

Le principali operazioni richieste al sistema sono:

* Inserimento di un nuovo partecipante
* Inserimento di una nuova prova
* Ricerca dei partecipanti
* Ricerche sulle prove
* Ricerca del migliore atleta di ogni categoria
* Operazioni di update e rimozione

Vediamo ora nello schema E/R, per ognuna di queste operazioni, i passi da compiere per ottenere il risultato desiderato.

*Inserimento di un nuovo partecipante*

Questa operazione inserisce un nuovo partecipante all’interno del Database, indipendentemente che esso sia un atleta, un giudice o un arbitro. Tuttavia vi è un vincolo legato all’esistenza della palestra dalla quale tale partecipante proviene: avendo la chiave importata dell’entità PALESTRA, ovvero la composizione del nome e della città in cui si trova, prima di poter inserire un nuovo record dobbiamo controllare l’esistenza di questa palestra nella Basi di Dati. Qualora non esistesse, viene richiesto all’Amministratore di sistema di inserire la suddetta palestra e reinserire il nuovo partecipante; viceversa, l’inserimento andrà a buon fine.

*Inserimento di una nuova prova*

La prova, come già sottolineato, è uno dei principali elementi del Database in quanto rappresenta l’esecuzione, da parte di un atleta, di un certo Kata.

Questa entità ha come identificatore la composizione del nome della prova, solitamente “kata” o “bunkai”, del nome del Kata eseguito e del Codice Fiscale dell’atleta.

Prima di inserire un nuovo record, anche qui, dobbiamo eseguire i controlli alla pressione del pulsante Aggiungi presente nella GUI, che manda l’informazione al Database: se il CF dell’atleta inserito esiste e il nome del Kata è presente nella Base di Dati, allora si è in grado di inserire il nuovo record. Altrimenti, nel caso in cui venisse a mancare la prima condizione l’Amministratore dovrà creare l’atleta che ha sostenuto la prova, se venisse a mancare la seconda egli dovrà inserire il Kata nella rispettiva tabella. Dopodiché potrà inserire la prova.

*Ricerca di un partecipante*

La ricerca di un partecipante si svolge sostanzialmente ricercando, tramite l’operazione di SELECT, il Codice Fiscale della persona e restituendo le sue anagrafiche e mostrandole nella GUI. Se il CF non esiste non dovrà essere visualizzato nulla.

*Ricerche sulle prove*

Sull’entità PROVA vengono eseguite varie operazioni di ricerca leggermente più complesse. Ad esempio vi è la possibilità di ricercare il migliore atleta per ogni categoria, sfruttando l’attributo “votazione” che è univocamente determinato dal nome della prova, dal CF dell’atleta e dal Kata che ha eseguito. Si ha anche la possibilità di ottenere l’anagrafica di tutti gli atleti che hanno eseguito una certa prova su uno specifico Tatami.

Altre operazioni le vedremo meglio, e nel dettaglio, successivamente.

Trasformazione dallo schema concettuale allo schema logico

In questa sezione vediamo come procedere per giungere alla rappresentazione sotto forma di schema logico del nostro E/R.

Innanzitutto, vediamo le chiavi importate e gli identificatori delle entità.

* **PARTECIPANTE**: essendo collegato a PALESTRA tramite un’associazione 1-N (in quanto un partecipante proviene da una ed una sola palestra mentre da una palestra possono provenire N atleti) questa entità dovrà possedere la chiave importata di PALESTRA. Al momento della trasformazione in schema logico eseguiremo un collasso verso il basso così da spostare gli attributi nelle sotto-entità della gerarchia ATLETA, ARBITRO e GIUDICE. ATLETA sarà collegato tramite un’associazione 1-N con CATEGORIA, per cui avremo come chiave importata anche il nome della categoria a cui appartiene.
* **PALESTRA**: una generica palestra è univocamente definita dalla composizione di “nome” e “città”, in quanto non possono esistere due palestre che hanno lo stesso nome nella medesima città.
* **GARA**: questa entità rappresenta il contesto fondamentalmente. Una gara si svolge in una palestra mentre in una palestra si possono svolgere N gare, di conseguenza importiamo la chiave di palestra in questa entità e come identificatore combineremo la chiave importata di palestra e la data della gara. Da questo deduciamo che in una certa palestra e in una stessa data, non si può svolgere più di una gara.
* **KATA**: un Kata è univocamente determinato dal suo nome. Non esistono più Kata con lo stesso nome.
* **CATEGORIA**: entità utilizzata per tenere la lista delle categorie. Ogni categoria è univocamente determinata dal suo nome.
* **PROVA**: entità che rappresenta l’esecuzione della prova da parte di un atleta. Essa è collegata tramite un’associazione 1-N con ATLETA e tramite un’associazione 1-N con Kata, conseguentemente importiamo le chiavi sia dell’una che dell’altra entità e definiamo l’identificatore di PROVA come la composizione del CF dell’atleta, del nome del Kata eseguito e del nome della prova.
* **TATAMI**: un tatami è univocamente determinato dal suo codice, ma essendo collegato a PALESTRA tramite un’associazione 1-N avrà la chiave importata di PALESTRA. E’ anche collegata tramite un’associazione 1-N a TAVOLO DI GIURIA, quindi avremo anche il codice del tavolo importato.
* **TAVOLO DI GIURIA**: un tavolo di giuria è univocamente determinato dal suo codice.

Eliminazione eventuali gerarchie

Dal momento che le entità ATLETA, ARBITRO e GIUDICE hanno i medesimi attributi (ad eccezione di ATLETA che possiede altri due campi), nello schema E/R le abbiamo messe come sotto-entità della gerarchia PARTECIPANTE. Ora, per passare allo schema logico, dobbiamo eseguire un collasso: dato che non possiamo utilizzare PARTECIPANTE in quanto un arbitro, ad esempio, non può essere un atleta, dobbiamo collassare verso il basso, spostando quindi ogni attributo dell’entità PARTECIPANTE nelle entità sottostanti.

Avremo quindi:

Traduzione delle entità

Traduciamo qui le entità del nostro Database. Il campo sottolineato corrisponde alla chiave primaria, mentre le chiavi importate sono precedute dal nome (in maiuscolo) dell’entità da cui derivano.

* ATLETA(CF, nome, cognome, data\_di\_nascita, luogo\_di\_nascita, sesso, grado\_Dan, PALESTRA: nomePalestra, PALESTRA: cittàPalestra)
* ARBITRO(CF, nome, cognome, data\_di\_nascita, luogo\_di\_nascita, PALESTRA: nomePalestra, PALESTRA: cittàPalestra)
* GIUDICE(CF, nome, cognome, data\_di\_nascita, luogo\_di\_nascita, PALESTRA:nomePalestra, PALESTRA:cittàPalestra)
* KATA(nome\_Kata)
* CATEGORIA(nome\_categoria)
* TATAMI(codice, TAVOLO\_DI\_GIURIA:CodiceTavoloGiuria, PALESTRA:nomePalestra, PALESTRA:cittàPalestra)
* TAVOLO\_DI\_GIURIA(codice)
* GARA(data, PALESTRA:nomePalestra, PALESTRA:cittàPalestra, nome\_manifestazione)
* PALESTRA(nome, città, indirizzo, numero\_posti\_spettatori)
* PROVA(nomeProva, nomeKata, ATLETA:CodiceFiscaleAtleta, categoria, ARBITRO:CodiceFiscaleArbitro, TATAMI:codiceTatami, votazione)

 Traduzione delle associazioni

Traduciamo le associazioni tra le entità.

* atleta\_proviene(CodiceFiscaleAtleta, nomePalestra, cittàPalestra

 Associazione che lega ATLETA e PALESTRA. Un atleta può provenire da una ed

 una sola palestra mentre da una palestra possono provenire N atleti.

* arbitro\_proviene(CodiceFiscaleArbitro, nomePalestra, cittàPalestra)

 Associazione che lega ARBITRO e PALESTRA. Un arbitro può provenire da una ed una sola palestra mentre da una palestra possono provenire N arbitri.

* giudice\_proviene(CodiceFiscaleGiudice, nomePalestra, cittàPalestra)

 Associazione che lega GIUDICE e PALESTRA. Un giudice può provenire da una

 ed una sola palestra mentre da una palestra possono provenire N giudici.

* ospitare(data, nomePalestra, cittàPalestra)

 Associazione che lega GARA e PALESTRA. Una gara può essere ospitata da una sola palestra mentre una palestra può ospitare N gare.

* eseguire(CodiceFiscaleAtleta, nomeProva, nomeKata)

 Associazione che lega ATLETA e PROVA.

* prevedere(nomeKata, CodiceFiscaleAtleta, nomeProva)

 Associazione che lega PROVA e KATA. Una prova prevede l’esecuzione di un Kata mentre un Kata può essere eseguito in N prove.

* appartenere(CodiceFiscaleAtleta, nome\_categoria)

 Associazione che lega ATLETA e CATEGORIA. Un atleta appartiene ad una ed una sola categoria, mentre ad una categoria possono appartenere N atleti.

* arbitrare(nomeKata, CodiceFiscaleAtleta, nomeProva, CodiceFiscaleArbitro)

 Associazione che lega PROVA e ARBITRO. Una prova viene giudicata da 5 arbitri mentre un arbitro può arbitrare N prove.

* comporre(CodiceFiscaleGiudice, CodiceTavoloGiuria)

 Associazione che lega GIUDICE e TAVOLO\_DI\_GIURIA. Dato che l’associazione

 è N-N, al momento della realizzazione dello schema logico questa diventerà un’entità con chiave composta (CodiceFiscaleGiudice, CodiceTavoloGiuria).

* disporre(CodiceTatami, nomePalestra, cittàPalestra)

 Associazione che lega TATAMI e PALESTRA. Un tatami appartiene ad una ed una sola palestra mentre ad una palestra appartengono N tatami.

* esecuzione(nomeProva, nomeKata, CodiceFiscaleAtleta, codiceTatami)

 Associazione che lega PROVA e TATAMI. Su un tatami si possono eseguire N prove, mentre una prova si esegue su un solo tatami.

*In conclusione, come tabelle, avremo:*

* ATLETA(CF, nome, cognome, data\_di\_nascita, luogo\_di\_nascita, sesso, grado\_Dan, PALESTRA: nomePalestra, PALESTRA: cittàPalestra)
* ARBITRO(CF, nome, cognome, data\_di\_nascita, luogo\_di\_nascita, PALESTRA: nomePalestra, PALESTRA: cittàPalestra)
* GIUDICE(CF, nome, cognome, data\_di\_nascita, luogo\_di\_nascita, PALESTRA:nomePalestra, PALESTRA:cittàPalestra)
* KATA(nome\_Kata)
* CATEGORIA(nome\_categoria)
* TATAMI(codice, TAVOLO\_DI\_GIURIA:CodiceTavoloGiuria, PALESTRA:nomePalestra, PALESTRA:cittàPalestra)
* TAVOLO\_DI\_GIURIA(codice)
* GARA(data, PALESTRA:nomePalestra, PALESTRA:cittàPalestra, nome\_manifestazione)
* PALESTRA(nome, città, indirizzo, numero\_posti\_spettatori)
* PROVA(nomeProva, nomeKata, ATLETA:CodiceFiscaleAtleta, categoria, ARBITRO:CodiceFiscaleArbitro, TATAMI:codiceTatami, votazione)
* GIUDICE\_TAVOLODIGIURIA(GIUDICE:CodiceFiscaleGiudice, TAVOLO\_DI\_GIURIA:CodiceTavoloGiuria)

Query SQL implementate

*Atleti:*

1) Numero di atleti appartenenti ad una certa categoria:
 SELECT nomeCategoria, COUNT(\*)

 FROM athletTable

 WHERE nomeCategoria = inputCategory

2) Dati di un atleta dato il suo Codice Fiscale:

 SELECT \*

 FROM athletTable

 WHERE CF = inputCF

3) Lista di atleti provenienti da una certa palestra e con un certo nome:

 SELECT \*

 FROM athletTable

 WHERE nome = inputName

 AND nomePalestra = inputGymName

 AND cittàPalestra = inputGymCity

*Palestra:*1) Nomi palestre in una certa città:

 SELECT DISTINCT nome

 FROM GymTable

 WHERE città = inputCity

2) Dettagli palestra dati città e nome:

 SELECT \*

 FROM GymTable

 WHERE nome = inputGymName

 AND città= inputGymCity

*Gara:*

1. Dettagli gara avendo la data e il nome della manifestazione:

 SELECT \* FROM CompetitionTable

 WHERE data = inputDate

 AND nome\_manifestazione= inputCompetitionName

*Prova:*

1. Anagrafica di tutti atleti che hanno eseguito una prova su un determinato Tatami:

 SELECT \*

 FROM atleta

 WHERE CF IN (

 SELECT CodiceFiscaleAtleta

 FROM prova p, atleta

 WHERE p.CodiceFiscaleAtleta = a.CF

 AND p.CodiceTatami = inputTatamiCode

 )

2) Somma dei punteggi ottenuti da un certo atleta:

 SELECT SUM(votazione) AS sumVote

 FROM prova

 WHERE CodiceFiscaleAtleta = inputAthletCF

3) Atleta che ha realizzato il punteggio massimo:

 SELECT a.CF, p.votazione

 FROM atleta a, prova p

 WHERE a.CF = p.CodiceFiscaleAtleta

 AND p.votazione = (

 SELECT MAX(votazione) as maxVote
 FROM atleta a2, prova p2

 WHERE a2.CF = p2.CodiceFiscaleAtleta

 )