

Database Exercise System

By Stefano Borzi & Alessandro Maggio

Sommario

Come usare il *Database Exercise System*

- Lato Utente
- Lato Admin
- Come funziona

Analisi dei requisiti

- Dominio del problema
- Operazioni
- Glossario

Progettazione concettuale

- Schema ER

Progettazione Logica

- Schema Relazionale
- Modello Relazionale

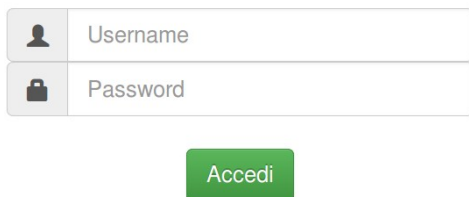
Progettazione Fisica

- Struttura del database (SQL)

Come usare il *Database Exercise System*

La parte amministrativa di questo sistema d'esercitazione è strutturata principalmente in due pagine raggiungibili dai pulsanti rossi presenti nell'header.

Entrambe le pagine sono protette da *user-name* e *password* per evitare che utenti non autorizzati apportino modifiche ai database presenti e/o alle domande, categorie etc.



The login form consists of two input fields stacked vertically. The top field is labeled 'Username' and has a person icon on the left. The bottom field is labeled 'Password' and has a lock icon on the left. Below these fields is a green button labeled 'Accedi'.

Il pannello di login contiene semplicemente un pulsante di conferma e due *input-box* che richiedono i dati per l'autenticazione.

La prima pagina che esamineremo sarà quella chiamata “*dbmanager*” la quale permetterà all'amministratore di creare dei nuovi database fittizi.

Le prime informazioni richieste sono il *nome del database* e il *numero di tabelle* presenti in esso.

Inserisci il nome del database:

molo|

Inserisci il numero di tabelle:

1

La modifica del secondo *input-box* (con valore maggiore e/o uguale a uno) mostrerà automaticamente lo schema di creazione delle singole tabelle, esso conterrà nella sua parte sinistra tutte le proprietà modificabili della tabella in questione e nella parte destra visualizzerà dinamicamente la tabella stessa.

In particolare le proprietà richieste per ogni singola tabella sono:

- *Nome tabella.*
- *Numero di attributi (colonne).*
- *Nomi dei singoli attributi.*
- *Numero di righe.*

Prendendo come esempio la tabella “*Marinaio*” e popolando tutti i campi vedremo a video l'esempio sottostante.

Tabella numero 1 (Marinaio)

Inserisci il nome della tabella:

Marinaio

Inserisci il numero di attributi (colonne):

4

Attr 1: idm
Attr 3: rating

Attr 2: nomem
Attr 4: eta

Inserisci il numero di valori (righe):

4

#	idm	nomem	rating	eta
1	134 FK	Aldo FK	8 FK	40 FK
2	523 FK	Mario FK	6 FK	25 FK
3	30 FK	Alessandro FK	5 FK	30 FK
4	4 FK	Giovanni FK	9 FK	19 FK

Per ogni tabella inoltre sono presenti delle piccole impostazione raggiungibili dal pulsante (rappresentato dalla chiave inglese) posto prima della scritta “*Tabella numero n*”.

Questo mini-pannello permette “*l'autoload*” e l'abilitazione di *chiave esterne*.

Impostazioni Tabella

idm
AutoLoad
☐ FK
rating
AutoLoad
☐ FK

nomem
AutoLoad
☐ FK
eta
AutoLoad
☐ FK

Nome
Cognome
Colore
Città
Saldo

OK

L'*autoload* non fa altro che aiutare l'amministratore nel popolare automaticamente i vari campi utilizzando valori predefiniti contenuti nelle varie categorie selezionabili.

Se la chiave esterna è stata abilitata dal pannello di impostazioni vedremo che tutti gli *input-box* del campo in questione si disabiliteranno e diventeranno attivi i vari pulsanti *FK* che permetteranno di definire le varie chiavi esterne.

#	idm		idb		data	
1	134	FK		FK		FK
2		FK		FK		FK

Se la chiave esterna non è mai definita alla pressione del tasto *FK* si aprirà un pannello che permetterà di selezionare la *tabella*, l'*attributo* e il *valore* da collegare.

Foreign Key

Seleziona una tabella:

1 - Marinaio

Seleziona un attributo:

idm

Seleziona il valore :

134
523
30
4

K Cancel

Foreign Key

Seleziona il valore :

30
134
523
30
4

K Cancel

Se invece il campo ha già un attributo e una tabella di riferimento verrà chiesto solo il valore.

L'utente amministratore inoltre sarà aiutato dagli *alert* che controlleranno la corretta creazione del nuovo database.

Per confermare il tutto è necessario cliccare sul tasto “*Aggiungi un nuovo database*”.

Aggiungi un nuovo database

Attenzione! Assicurati di aver riempito tutti i campi.

Attenzione! Le tabelle non possono avere nome nullo o uguale.

Attenzione! Non è possibile avere attributi nulli o con lo stesso nome nella stessa tabella.

La seconda pagina di amministrazione invece permette di inserire nuove domande e/o aggiungere nuovi argomenti e soluzione

Analisi dei requisiti

Dominio del problema

Nel dipartimento di Matematica e Informatica (DMI) gli studenti che seguono la materia Database non utilizzano e/o non hanno uno strumento adatto per esercitarsi con l' Algebra Relazionale e l' SQL.

A tal proposito diversi studenti come progetto della materia hanno intenzione di creare uno strumento di esercitazione che colmi questa mancanza.

Questo sistema di esercitazione dovrà però, fornire oltre che ad un'interfaccia per gli studenti dove poter eseguire le proprie query e visualizzare il risultato e la soluzione dell'esercizio (ed il risultato della soluzione-query dell'esercizio) anche un pannello di amministrazione in modo da poter far aggiornare al docente gli esercizi del sistema di esercitazione.

Operazioni

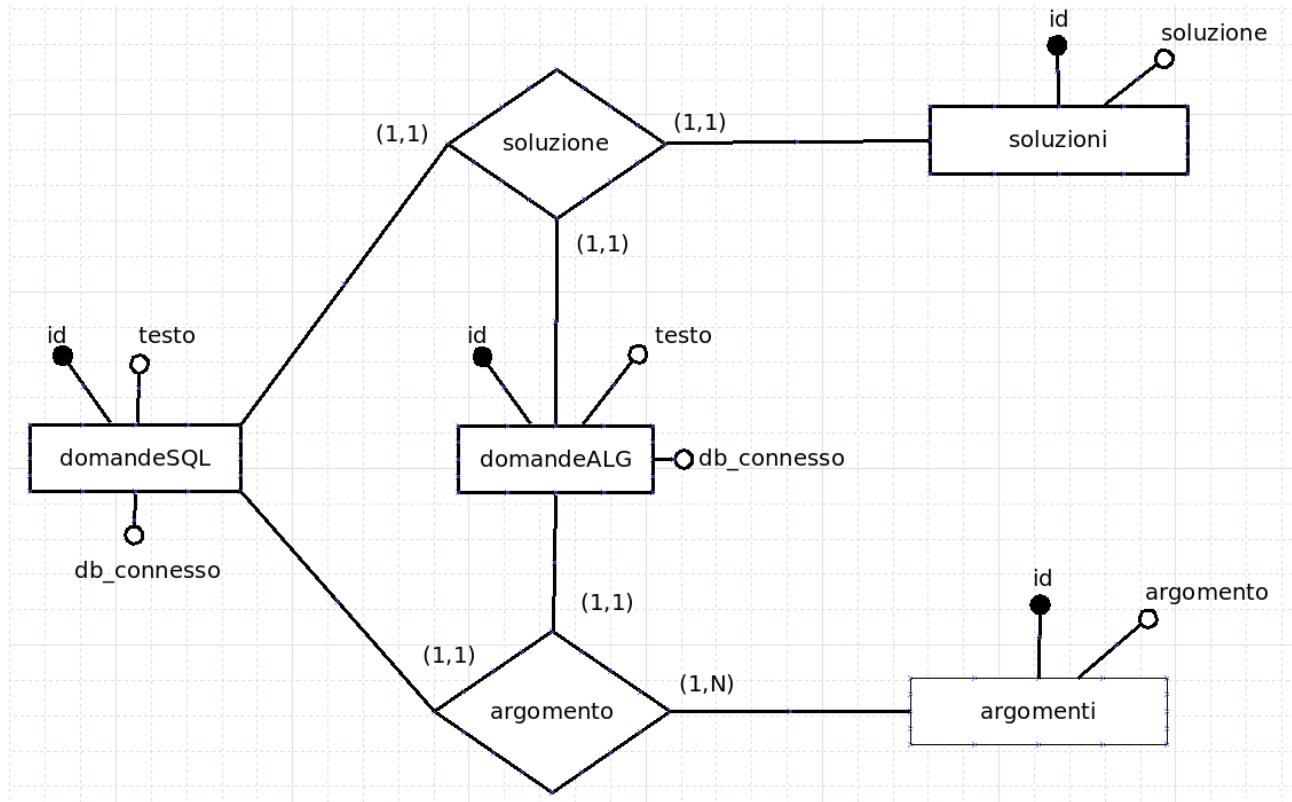
1. Carico di nuovi esercizi e nuovi database relativi agli esercizi (6-10 volte all'anno)
2. Query “SELECT” degli esercizi (frequenza arbitraria a seconda del numero degli studenti che utilizzano il sistema)

Glossario

Termine	Descrizione	Collegamenti
Docente	Utente che utilizzerà il pannello di amministrazione per poter aggiungere esercizi e “database” relativi agli esercizi	DomandeSQL/ALG, soluzioni, argomenti, <i>possibilità di inserire nuovi “database”</i>
Studente	Utente che svolgerà gli esercizi caricati nel database	Database relativi agli esercizi

Progettazione concettuale

Schema ER



Progettazione logica

Schema relazionale

soluzioni (id, soluzione)

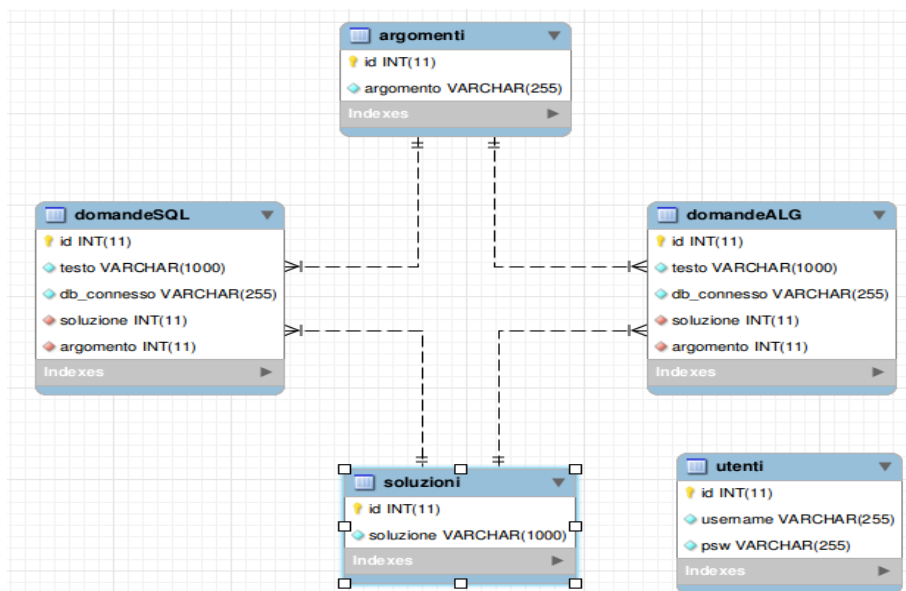
argomenti(id, argomento)

domandeSQL (id, testo, db_connesso, soluzione, argomento)

domandeALG (id, testo, db_connesso, soluzione, argomento)

utenti(id, username, psw)

Modello relazionale



Progettazione Fisica

-- Create 'argomenti' table

```
CREATE TABLE IF NOT EXISTS argomenti(  
  id INT PRIMARY KEY AUTO_INCREMENT NOT NULL,  
  argomento varchar(255) NOT NULL  
);
```

-- Create 'soluzioni' table

```
CREATE TABLE IF NOT EXISTS soluzioni(  
  id INT PRIMARY KEY AUTO_INCREMENT NOT NULL,  
  soluzione varchar(1000) NOT NULL  
);
```

-- Create 'domandeALG' table

```
CREATE TABLE IF NOT EXISTS domandeALG(  
  id INT PRIMARY KEY AUTO_INCREMENT NOT NULL,  
  testo varchar(1000) NOT NULL,  
  db_connesso varchar(255) NOT NULL,  
  soluzione INT NOT NULL,  
  argomento INT NOT NULL,  
  FOREIGN KEY (soluzione) REFERENCES soluzioni(id),  
  FOREIGN KEY (argomento) REFERENCES argomenti(id)  
);
```

-- Create 'domandeSQL' table

```
CREATE TABLE IF NOT EXISTS domandeSQL(  
  id INT PRIMARY KEY AUTO_INCREMENT NOT NULL,  
  testo varchar(1000) NOT NULL,  
  db_connesso varchar(255) NOT NULL,  
  soluzione INT NOT NULL,  
  argomento INT NOT NULL,  
  FOREIGN KEY (soluzione) REFERENCES soluzioni(id),  
  FOREIGN KEY (argomento) REFERENCES argomenti(id)  
);
```

```
CREATE TABLE IF NOT EXISTS utenti(  
  id INT PRIMARY KEY AUTO_INCREMENT NOT NULL,  
  username varchar(255) NOT NULL,  
  psw varchar(255) NOT NULL  
);
```