*Verifica di Informatica*

05/04/2018 Caruso Denny VB

**Analisi dei dati:**

Il problema chiede di realizzare una web chat attraverso l’utilizzo di un database mysql. La chat deve prevedere il login previa registrazione, ottenuto il quale l'utente loggato entrerà nella lista degli utenti online. La chat principale dovrà contenere tutti i messaggi digitati in modo cronologico dai vari utenti del tipo:

nomeutente: 25.04.2017 - 18.15.05

ciao a tutti sono Carlo

e dovrà essere visualizzata ogni tot ms in modo da visualizzare gli ultimi messaggi inviati. Inoltre ci viene anche chiesto di implementare opzionalente le seguenti funzionalità:

- CREARE uno spazio per le varie stanze della chat

- click su un utente online, apre una conversazione privata con l'utente cliccato

Per risolvere il problema proposto è necessario tener conto di almeno due entità da implementare nel modello E/R, poi passare alla costruzione del database con MySqli in tutti i suoi dettagli (rispettando ciò che viene descritto nel modello E/R) e infine scrivere pagine html e php per rendere possibile la connessione, l’utilizzo effettivo della chat e altri servizi elencati successivamente. Aggiungo inoltre fogli di stile CSS per migliorare l’asetto grafico del lavoro. Il problema richiede almeno le seguenti pagine:

* Homepage
* Form Registrazione Utente
* Reset Password
* Login
* Chat

Il linguaggio PHP oltre ad occuparsi della connessione col database, l’inserimento, la modifica e la verifica delle informazioni all’interno del database, si occuperà anche di gestire l’esatto riscontro dell’username dell’account in fase di registrazione per evitare duplicati. Inizio elencando le varie entità trovate e successivamente elencherò i dettagli di ciascuna di esse. Le entità che possono essere individuate nel problema sono:

* Utente, per rappresentare le informazioni riguardanti gli utenti che si registrano e che inviano messaggi nella chat. Gli attributi di questa entità sono: username (chiave primaria che permette di identificare univocamente un utente), passcode (password dell’utente per accedere alla chat), email (scelta dall’utente per usufruire di servizi come recupero della password), nome (nome proprio dell’utente), cognome (conome proprio dell’utente) e datanascita (data di nascita dell’utente registratosi).
* Messaggio, per rappresentare le informazioni riguardanti i Messaggi inviati all’interno della chat. Gli attributi di questa entità sono: codMessaggio (codice identificativo di ogni singolo messaggio inviato all’interno della chat), datainvio (per contenere i dati relativi alla data e all’orario di invio del messaggio all’interno della chat), messaggio (contenente il testo stesso del messaggio) e username (in qualità di chiave esterna per associare un messaggio inviato a un utente in particolare).

L’associazione ricavata dal problema è:

* Invia: fra Utente e Messaggio esiste una correlazione 1 a N: Un utente può inviare molti (uno o più) Messaggi. Dal verso opposto, un Messaggio può essere inviato da uno e un solo utente (in quanto quel messaggio con quel particolare codice identificativo viene inviato da un solo utente e non da più utenti nonostante il fatto che tutti gli utenti possano inviare un messaggio). L’associazione non presenta attributi.

**Progettazione**

Una volta creata la homepage e le successive pagine (login, form di registrazione, recupera password e così via) elaborate in HTML (HyperText Markup Language (HTML; traduzione letterale: linguaggio a marcatori per ipertesti) è un linguaggio di markup. Nato per la formattazione e impaginazione di documenti ipertestuali disponibili nel web 1.0, oggi è utilizzato principalmente per la rappresentazione della struttura logica di una pagina web (definita appunto dal markup) e la sua rappresentazione, gestita tramite gli stili CSS per adattarsi alle nuove esigenze di comunicazione e pubblicazione all'interno di Internet.

L'HTML è un linguaggio di pubblico dominio, la cui sintassi è stabilita dal World Wide Web Consortium (W3C). È derivato dall'SGML, un metalinguaggio finalizzato alla definizione di linguaggi utilizzabili per la stesura di documenti destinati alla trasmissione in formato elettronico. La versione attuale, la quinta, è stata rilasciata dal W3C nell'ottobre 2014.

Invece per quanto riguarda la creazione del database, questa viene fatta mediante lo strumento MySQL: un Relational database management system (RDBMS) composto da un client a riga di comando e un server. Entrambi i software sono disponibili sia per sistemi Unix che per Windows. Il software ci viene fornito insieme a PHPmyAdmin un'applicazione web scritta in PHP, distribuita con licenza GPL, che consente di amministrare un database MySQL o MariaDB tramite un qualsiasi browser. L'applicazione è indirizzata sia agli amministratori del database, sia agli utenti. Gestisce i permessi prelevandoli dal database. Inoltre phpMyAdmin permette di creare un database da zero, creare le tabelle ed eseguire operazioni di ottimizzazione sulle stesse. Presenta un feedback sulla creazione delle tabelle per evitare eventuali errori. Sono previste delle funzionalità per l'inserimento dei dati (popolazione del database), per le query, per il backup dei dati, e così via.

Questi strumenti (phpMyAdmin e MySQL) ci vengono forniti da un unico software: UwAmp che permette appunto di trasformare Windows (o comunque il localhost) in un server vero e proprio per elaborare e visualizzare pagine PHP.

L’ultimo strumento utilizzato è per l’appunto PHP, un linguaggio di scripting interpretato, originariamente concepito per la programmazione di pagine web dinamiche. Viene utilizzato per la back-end o server-side di un sito, per la sua gestione e organizzazione in tutti i suoi aspetti; ma può essere usato anche per scrivere script a riga di comando o applicazioni stand-alone con interfaccia grafica. Non ho considerato gli aspetti opzionali, in quanto tali per semplificare il carico di lavoro. Tali aspetti potranno essere implementati successivamente.

Di seguito riporto il relativo dizionario delle entità e successivamente quello delle associazioni:

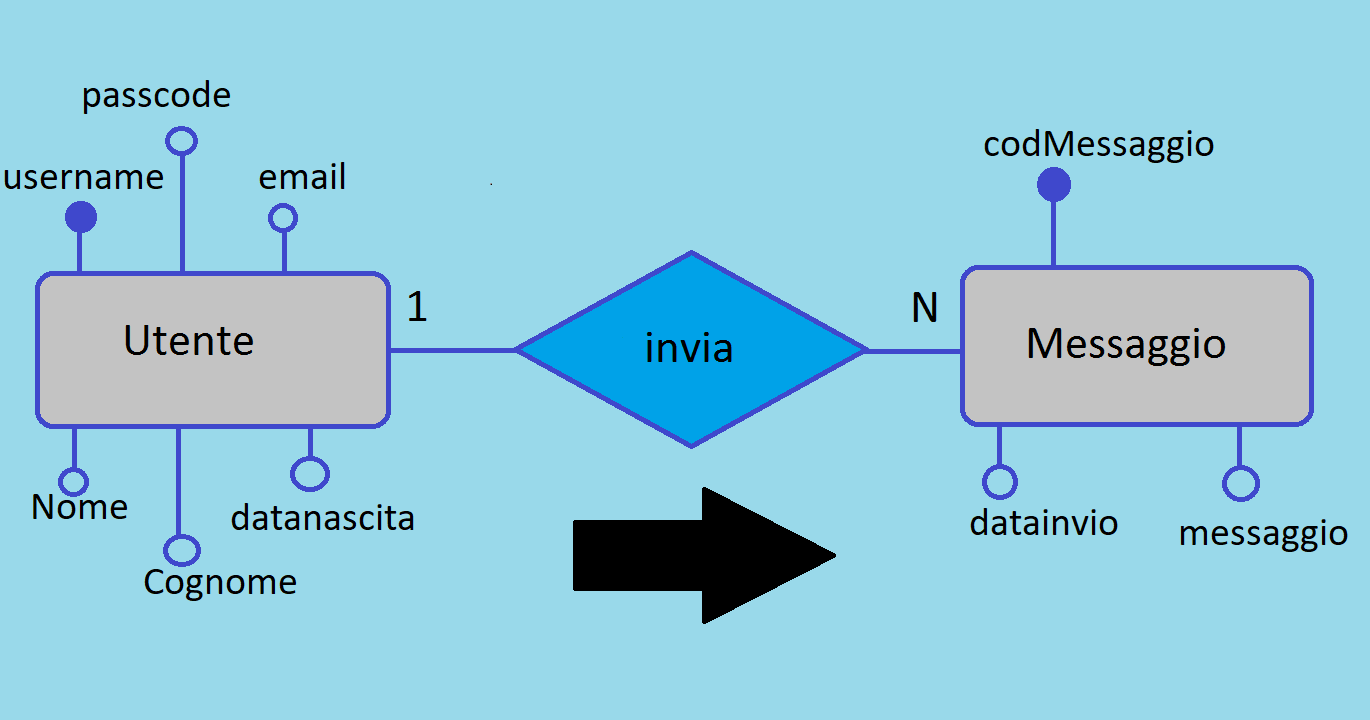
**Dizionario Entità**

|  |  |  |  |
| --- | --- | --- | --- |
| Nome Entità | Descrizione | Attributi | Identificatore |
| Utente | Rappresenta le informazioni riguardanti tutti gli Utenti. | +username  +passcode  +email  +nome  +cognome  +datanascita | username |
| Messaggio | Rappresenta le informazioni riguardanti tutti i Messaggi. | +codMessaggio  +datainvio  +messaggio | codMessaggio |

**Dizionario Associazioni**

|  |  |  |  |
| --- | --- | --- | --- |
| Nome Entità | Descrizione | Entità coinvolte | Attributi |
| Invia | Associa Utente a Messaggio | Utente (1)  Messaggio (N) |  |

**Modello E/R**

****

**Regole di Derivazione**

**Utente invia Messaggio (1:N)**

Utente:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nome | Chiave | Descrizione | Tipo | Vincolo |
| username | PK | Username che identifica univocamente un utente all’interno del sistema. | Testo | Obbligatorio |
| passcode |  | Indica la password dell’utente per accedere alla chat. | Testo | Obbligatorio |
| email |  | Email utilizzata dall’utente in fase di registrazione per eventuali contatti. | Testo | Obbligatorio |
| nome |  | Nome proprio dell’utente. | Testo | Obbligatorio |
| cognome |  | Cognome proprio dell’utente. | Testo | Obbligatorio |
| datanascita |  | Data di nascita dell’utente. | Data | Obbligatorio |

Messaggio:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nome | Chiave | Descrizione | Tipo | Vincolo |
| codMessaggio | PK | Codice che identifica univocamente un messaggio. | Testo | Obbligatorio |
| datainvio |  | Indica la data di invio del messaggio. | Data | Obbligatorio |
| messaggio |  | Messaggio di testo inviato dall’utente. | Testo | Obbligatorio |
| username | FK | Chiave esterna per creare associazione con l’entità utente. | Testo | Obbligatorio |

**Query**

Create Table Utente:

CREATE TABLE utente (

username varchar(25) PRIMARY KEY UNIQUE AUTO\_INCREMENT NOT NULL,

passcode varchar(25) NOT NULL,

email varchar(50) NOT NULL,

nome varchar(30) NOT NULL,

cognome varchar(30) NOT NULL,

datanascita date NOT NULL);

Create Table Messaggio:

CREATE TABLE messaggio (

codMessaggio int(50) PRIMARY KEY UNIQUE AUTO\_INCREMENT NOT NULL ,

datainvio datetime NOT NULL,

messaggio varchar(3000) NOT NULL,

username varchar(25) FOREIGN KEY REFERENCES utente(username));

Qui di seguito riporto inoltre le query di inserimento di un utente e di un messaggio all’interno del database:

Insert Into Utente:

$agg = "INSERT INTO chat.utente(username, passcode, email, nome, cognome, datanascita) values ('$username', '$passcode', '$email','$nome','$cognome', '$datanascita')";

Insert Into Messaggio:

$agg = "INSERT INTO chat.messaggio(codMessaggio, datainvio, messaggio, username) values (NULL, '$datainvio', '$messaggio','$username')";

**Sviluppo**

La gestione della chat è stata particolarmente divertente nella creazione di una pagina dedicata alla chat vera e propria che contenesse tutti i messaggi inviati dagli utenti (anche in contemporanea) e che visualizzasse tutti gli utenti al momento online (live sempre in contemporanea). L’idea è nata dal congiungere più pagine PHP attraverso un’unica sessione di lavoro dedicata che permetteva alle pagine di scambiarsi i dati principali degli utenti provenienti dal database come username e stato (0=offline, 1=online).

La gestione online/offline infatti è stata gestita proprio in questo modo ovvero apponendo un altro campo nel database di tipo BIT che contenesse 0 quando l’utente è offline o effettua il logout oppure 1 quando questi invece effettua il login e quindi è attivo nella chat. Ho inoltre inserito le modifiche necessarie affinchè alla chiusura della scheda l’utente non fosse scollegato automaticamente, ma rimane attualmente collegato in modo da riprendere la conversazione successivamente senza dover per forza inserire di nuovo tutti i dati necessari per effettuare il login. La strutturazione della chat a livello grafico è stata ottenuta mediante l’implementazione di molte textarea e tabelle tra di loro, incastrate una nell’altra e anche responsive fino a un certo punto. L’aspetto grafico è stato conferitogli anche grazie a fogli di stile CSS e grazie all’uso di tag in-line per la modifica immediata dell’aspetto della pagine web.

Qui di seguito riporto il codice PHP per connettersi al database:

<?php

session\_start();

$conn = mysqli\_connect("localhost", "root", "root","chat");

if(!$conn)

{

echo("Errore di connessione.");

exit();

}

?>

Qui di seguito invece riporto il codice PHP per effettuare rispettivamente LOGIN e LOGOUT:

Login

if($\_SERVER["REQUEST\_METHOD"] == "POST") {

$myusername = mysqli\_real\_escape\_string($conn, $\_POST['username']);

$mypassword = mysqli\_real\_escape\_string($conn, $\_POST['password']);

$\_SESSION['user'] = $myusername;

$sql = "SELECT username FROM chat.utente WHERE username = '$myusername' and passcode = '$mypassword'";

$result = mysqli\_query($conn,$sql);

$row = mysqli\_fetch\_array($result,MYSQLI\_ASSOC);

$active = $row['username'];

$count = mysqli\_num\_rows($result);

if($count == 1) {

$\_SESSION['login\_user'] = $myusername;

echo("<h2>Username e Password Corrette. A breve sarai reindirizzato alla pagina per la chat.</h2>");

$agg = "UPDATE utente SET status = '1' WHERE username='$myusername'";

$result = mysqli\_query($conn,$agg);

header( "refresh:1;url=chat.php" );

Logout

$myusername=$\_SESSION['user'];

$agg = "UPDATE utente SET status = '0' WHERE username='$myusername'";

$result = mysqli\_query($conn,$agg);

session\_destroy();

header('Location: login.html');

exit;

Qui di seguito per esempio riporto il codice css per la realizzazione dei simpatici bottoni all’interno delle pagine web:

.myButton {

background: #5e7689;

background-image: -webkit-linear-gradient(top, #5e7685, #1fa0f0);

background-image: -moz-linear-gradient(top, #5e7685, #1fa0f0);

background-image: -ms-linear-gradient(top, #5e7685, #1fa0f0);

background-image: -o-linear-gradient(top, #5e7685, #1fa0f0);

background-image: linear-gradient(to bottom, #5e7685, #1fa0f0);

-webkit-border-radius: 60;

-moz-border-radius: 60;

border-radius: 60px;

font-family: Georgia;

color: #fffcff;

font-size: 20px;

padding: 10px 20px 10px 20px;

border: solid #000000 3px;

margin: 5px;

}

*Caruso Denny*

*VB*

*05/04/2018*