

Tema

Oggi la realizzazione di un sito web è agevolata dalla presenza di software caratterizzati da un'interfaccia utente relativamente semplice e, in certi casi, sono sufficienti pochi clic per realizzare il proprio spazio in rete, ma si pongono nuove problematiche relative ai requisiti e ai criteri di valutazione di questi software.

Svolgimento del tema

Da quando esistono i linguaggi di programmazione lato server sono state realizzate applicazioni che aiutano il webmaster a realizzare un sito web. Alcuni esempi di questo tipo di software sono: Wordpress, Joomla, Drupal e MediaWiki. Tutti questi software sono stati sviluppati da una comunità di programmatori che hanno rilasciato il codice sorgente con licenza libera oppure open source. Grazie alla libertà offerta da questi software sono nate spontaneamente molte altre librerie o plugin che li rendono sempre più semplici da usare e da personalizzare.

Questi software permettono anche all'utente, che naviga il sito web, di contribuire ai contenuti del sito (come nella famosa wikipedia).

La loro realizzazione nasconde una certa complessità e il loro uso aumenta i rischi di sicurezza intrinseci di tutte le applicazioni web lato server, e devono essere continuamente aggiornati.

Esistono anche applicazioni simili per i campi dell'e-commerce o della gestione aziendale (ERP), che possono essere distribuite anche con licenza proprietaria.

In ogni caso, i criteri di scelta del progettista che decide di utilizzare tali software, dovrebbero mirare, oltre alla riduzione dei costi, anche ad ottenere programmi che:

- producano dati in formato facilmente interscambiabile,
- che garantiscano degli aggiornamenti costanti,
- abbiano un buon supporto tecnico ad un prezzo accettabile
- utilizzino al meglio l'hardware (requisiti hardware)
- realizzino un sito facilmente accessibile.

La riduzione dei costi è evidente per i software open source, ma questi possono avere meno supporto tecnico rispetto ai software proprietari.

Il formato dei dati è importante per poter recuperare facilmente i dati anche a distanza di tempo.

I dati vengono quasi sempre salvati in un server DBMS e il progettista deve scegliere il software anche in base al DBMS più adatto alle proprie esigenze.

L'accessibilità è una funzione importante per le aziende e per le pubbliche amministrazioni, e può essere valutata in base al rispetto degli standard del web (html5, css3).

Vocabolario dei termini

dipendente: si devono registrare dati anagrafici, come nome, cognome, data di nascita, CF. Ogni dipendente può effettuare presenze con il proprio badge. Il badge è solo un codice che identifica il dipendente.

presenza: la presenza è costituita da un ingresso e un'uscita (data e ora). La presenza deve corrispondere ad un solo dipendente.

NOTA: Qualche volta l'orario dei dipendenti può essere flessibile e consentire più ingressi e più uscite, ma nel testo non è specificato nulla. Se fosse fatta tale ipotesi, si dovrebbero associare più ingressi e uscite ad ogni presenza giornaliera

assenza: per determinare l'assenza di un dipendente è necessario conoscere quali siano i giorni lavorativi durante l'anno e l'orario di lavoro dei dipendenti. Ad esempio, si può ipotizzare dal lunedì al venerdì. In un caso più realistico sarebbe necessaria una tabella dei giorni di chiusura.

ritardo: I ritardi del dipendente sono segnalati registrando e calcolando l'ora di ingresso. Una funzione calcolerà i ritardi, ma non è sviluppata perché non è richiesto. Per determinare il ritardo, oltre al giorno è necessario l'orario di lavoro, cioè conoscere i turni dei dipendenti. Anche qui, semplificando, si può ipotizzare che tutti i dipendenti lavorano dalle 8:00 alle 16:00.

NOTA: per la comunicazione automatica del ritardo o dell'assenza si dovrà realizzare una funzione che visualizza un messaggio di output, ma non è sviluppata perché non è richiesto.

ferie: è la richiesta di ferie di un dipendente: ha una data di richiesta, una data di inizio e di fine, e l'eventuale approvazione (vero o falso).

NOTA: Per calcolare il numero di ferie di cui ha usufruito un dipendente si può usare un'interrogazione, ma nell'esercizio non è richiesto lo sviluppo di questa funzione. Lo stesso per i tempi di approvazione.

permesso: la richiesta di permesso di un dipendente. Deve avere la data di richiesta, la data del permesso, la motivazione e l'eventuale approvazione.

NOTA: Per calcolare il numero di permessi di cui ha usufruito un dipendente si può usare un'interrogazione, ma non ne è richiesto lo sviluppo. Lo stesso per i tempi di approvazione.

contratto: ogni cliente può avere contratto part-time o a tempo pieno (può essere un attributo con un valore vero/falso oppure un'entità "durata").

tipodicontratto: tipologia di contratti dei dipendenti. Può assumere solo tre valori: vi è un codice identificatore e una descrizione che può valere: tempo determinato, indeterminato, apprendistato. Ogni tipo di contratto può essere applicato anche a più clienti.

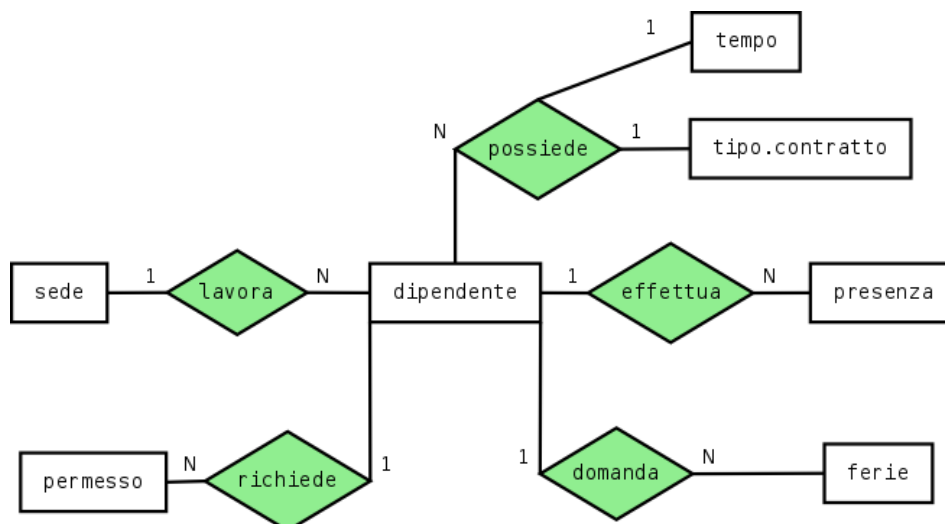
sede: la sede di lavoro del dipendente. Ha un codice identificativo, una via, un n. civico e una città. In una sede sono impiegati più dipendenti.

Schema E-R

Per completare lo schema dovrebbero essere aggiunti anche gli attributi sulle entità...

L'entità "tempo di impiego" rappresenta contratti part-time o tempo pieno. Nel caso di associazioni ternarie le regole di lettura sono un po' complicate. L'associazione ternaria diventa un modo per obbligare la presenza contemporanea di almeno un elemento appartenente a tutte e tre le entità.

Un'alternativa più semplice potrebbe essere quella di usare un semplice attributo VERO/FALSO sull'associazione.



Regole di lettura

Ogni dipendente possiede un tipo di contratto, con un tempo d'impiego

Ogni tipo di contratto, con un tempo d'impiego, è posseduto da uno o più dipendenti

Ogni tempo di impiego si applica ad un tipo di contratto e ad uno o più dipendenti

Ogni dipendente lavora in una sede

In ogni sede lavorano uno o più dipendenti

Ogni dipendente effettua una o più presenze

Ogni presenza è effettuata da un dipendente

Ogni dipendente richiede uno o più permessi

Ogni permesso è richiesto da un dipendente

Ogni dipendente domanda una o più ferie

Ogni ferie è domandata da un dipendente

Relazioni

sedi (idsede, via, civico, citta)

tipicontratto (idtipocontratto, descrizione)

durata (iddurata, descrizione)

dipendenti (cf, badge, nome, cognome, datanascita, luogonascita, part-time, idtipocontratto, idsede)

possessi (cf, idtipocontratto, iddurata)

presenze (idpresenza, ingresso, uscita, cf)

permessi (idpermesso, datarichiesta, data, motivazione, approvazione, dataapprovazione, cf)

ferie (idferie, datarichiesta, datainizio, datafine, approvazione, dataapprovazione, cf)

NOTA: La relazione "possessi" ha una chiave primaria composta da tre chiavi esterne

Testing: riempire sempre le tabelle per provarle...

Schema dei dati

Archivio	campo	vincolo	tipo di dato	dimensione	
sedi	idsede	c. primaria	CHAR	16	
	via		VARCHAR	50	
	civico		VARCHAR	10	
	citta		VARCHAR	50	
tipicontratto	idtipocontratto	c. primaria	CHAR	1	
	descrizione	not null	VARCHAR	60	
durata	iddurata	c. primaria	CHAR	1	
	descrizione	not null	VARCHAR	60	
dipendente	cf	c. primaria	CHAR	16	
	badge	not null unique	CHAR	16	
	nome	not null	VARCHAR	50	
	cognome	not null	VARCHAR	50	
	datanascita		DATE		
	luogonascita		VARCHAR	50	
	part-time		BOOLEAN		
	idsede	c. esterna	CHAR	3	
	idtipocontratto	c. esterna	CHAR	1	
possessi	idtipocontratto	c. esterna UNIQUE	CHAR	1	
	iddurata	c. esterna UNIQUE	CHAR	1	
	cf	c. esterna	CHAR	16	
presenze	idpresenza	c. primaria	CHAR	16	
	ingresso	not null	TIMESTAMP		
	uscita		TIMESTAMP		
	cf	c. esterna	CHAR	16	
permessi	idpermesso	c. primaria	CHAR	6	
	datarichiesta	not null	DATE		
	data	not null	DATE		
	motivazione		VARCHAR	250	
	approvazione		BOOLEAN		

Archivio	campo	vincolo	tipo di dato	dimensione	
	dataapprovazione		DATE		
	cf	c. esterna	CHAR	16	
ferie	idferie	c. primaria	CHAR	6	
	datarichiesta	not null	DATE		
	datainizio	not null	DATE		
	datafine	not null	DATE		
	approvazione		BOOLEAN		
	dataapprovazione		DATE		
	cf	c. esterna	CHAR	16	

Osservazioni: si potrebbe inserire il vincolo NOT NULL sia sulla colonna "ingresso" che sulla colonna "uscita" nella tabella "presenze"? Se si mettesse questo vincolo anche sulla colonna "uscita" sarebbe impossibile registrare l'ingresso, in quanto sarebbe richiesto di inserire contemporaneamente anche l'uscita.

Interrogazioni

Le interrogazioni parametriche in SQL andrebbero costruite ed eseguite all'interno di un linguaggio di programmazione come il PHP.

- Visualizzare i dati identificativi dei dipendenti per una determinata sede

```
SELECT cf
FROM dipendenti NATURAL JOIN sedi
WHERE sede=$idsede;
```

- Visualizzare il numero di dipendenti per ogni tipologia di contratto

```
SELECT descrizione, COUNT(*)
FROM dipendenti NATURAL JOIN tipicontratti
GROUP BY descrizione;
```

- Visualizzare i dati identificativi delle sedi dove vi siano meno di 10 dipendenti

```
SELECT sedi.idsede, COUNT(cf)
FROM dipendenti NATURAL JOIN sedi
GROUP BY sedi.idsede
HAVING COUNT(cf)<10;
```

Descrizione del software

Per la gestione dei dati si può utilizzare MySQL un DBMS server open source.

L'interfaccia utente può essere realizzata in HTML e PHP. Nel lato client si può usare un qualsiasi browser, nel lato server si può usare il web server Apache con interprete PHP.

Tutti questi programmi sono disponibili sia per sistema operativo Microsoft che per Linux, ma sarebbe meglio utilizzare un hosting di tipo Linux perché sono nativi di questo sistema operativo

Funzioni svolte dal sistema software

Per riassumere le funzioni svolte dal software potrebbe essere utile pensare ad un menù.

Il sito che dovrebbe consentire la gestione automatizzata delle presenze, della richiesta dei permessi e della segnalazione dei ritardi può essere utilizzato solo dal personale della segreteria e dai dipendenti per effettuare la domanda di ferie, perciò è necessario una funzione per il login. Solo le ultime tre funzioni sono disponibili anche per i dipendenti.

- dipendenti: inserimento, ricerca, modifica, eliminazione, scelta tipo di contratto, conteggio ferie, conteggio permessi, conteggio ritardi
- sedi: inserimento, ricerca, modifica, eliminazione, conteggio dipendenti
- login
- richieste di permesso
- richieste di ferie

Mappa del sito web

Progettare il grafo della mappa del sito e dei collegamenti tra le pagine.

Devono essere presenti le pagine che realizzano le voci delle funzioni del precedente menù.

Pagina web richiesta

```
<html>
  <head><title>appartamento</title>
</head>
  <body>
    <?php
      $dbms="mysql";
      $host="indirizzo.dbms.it";
      $dbname="agenzia";
      $user="mysql-banca";
      $password="abc123";
      $cf=$_POST['cf'];
      $db = new PDO($dbms.':host=".'.$host.'";dbname=".'.$dbname,$user,$password);

// -----interrogazione per i dati dell'appartamento-----
      $espressione = $db->query("SELECT * FROM dipendenti
                                WHERE cf='".$cf."'");

      $risultato = $espressione->fetchAll(PDO::FETCH_ASSOC);
      $c1=key($risultato); //chiave del primo elemento

      echo "\n<table>\n <tr>\n";
      foreach($risultato[$c1] as $c => $v)
      {
        echo "\t <th> ".$c." </th> \n";
      }
      echo " </tr>\n";

      foreach($risultato as $riga => $val)
      {
        echo "\n <tr>\n";
        foreach($val as $c => $v)
        {
          echo "\t <td> ".$v." </td> \n";
        }
        echo " </tr>\n";
      }
      echo "\n</table>\n";
    </body>
</html>
```