

Esame di stato Mercurio 2005 Palestra

Traccia

Telelavoro ed e-commerce, due mini-rivoluzioni, stentano a svilupparsi nel nostro Paese a differenza di quanto è avvenuto negli Stati Uniti e in altre parti d'Europa.

Il candidato, dopo aver brevemente illustrato le caratteristiche tecniche di tali processi, analizzi le ragioni del loro limitato utilizzo in Italia e sviluppi il seguente punto.

Una palestra vuole informatizzare la gestione dei corsi offerti alla propria clientela che si caratterizzano per:

- la presenza di istruttori specializzati in ciascun corso;
- il numero massimo di partecipanti variabile per ciascun corso;
- l'offerta dello stesso corso in diversi orari della giornata;
- la partecipazione al corso con idoneo certificato medico di durata annuale.

Organizzare un database in grado di fornire le seguenti informazioni:

1. elenco degli iscritti a ogni corso
2. numero dei posti liberi in ciascun corso
3. elenco dei corsi tenuti da ogni istruttore
4. elenco dei clienti con certificato medico scaduto

e, dopo aver

- descritto le caratteristiche dell'applicativo che si intende utilizzare,
- elaborato il progetto dell'applicazione complessiva, comprendente l'analisi, lo schema, la definizione dei dati e il menu di scelta,

realizzare almeno uno dei quattro punti sopra indicati utilizzando l'applicativo descritto.

Progettare, infine, la pagina web con la quale la palestra presenta alla clientela i corsi, le relative caratteristiche e gli orari.

Dati mancanti opportunamente scelti.

E-commerce

Il termine e-commerce indica "commercio elettronico" cioè la compravendita, il marketing e la fornitura di prodotti o servizi attraverso computer collegati in rete.

Sebbene la definizione copra un campo piuttosto ampio in genere si intende per e-commerce l'acquisto di beni e servizi attraverso il servizio World Wide Web della rete Internet ricorrendo a servizi sicuri come ad esempio l'HTTPS, una variante del protocollo http che crittografa i dati sensibili dei clienti contenuti nell'ordine di acquisto allo scopo di tutelare il consumatore; queste tecnologie rendono possibili i servizi di pagamento on-line, come le autorizzazioni per il pagamento con carta di credito.

Un altro elemento chiave per l'e-commerce è il web attivo; infatti per effettuare efficaci transazioni sono necessarie pagine dinamiche in grado di mostrare cataloghi che si aggiornano di frequente e di gestire personalizzazioni delle pagine come ad esempio il "carrello della spesa".

I punti critici dell'e-commerce che ne limitano la diffusione sono le truffe e la sicurezza. La cifratura protegge da accessi non autorizzati ma nuove tecniche come il 'phishing' aggirano queste barriere creando spesso diffidenza nell'utenza.

Telelavoro

Il Telelavoro è una modalità di lavoro grazie alla quale, impiegando infrastrutture telematiche ed informatiche, è possibile valicare i tradizionali confini fisici e logistici dell'ufficio.

Le principali categorie di telelavoro sono:

- Telelavoro Domiciliare o Homeworking: si tratta di un'attività lavorativa svolta in prevalenza presso l'abitazione del lavoratore che deve essere dotata di una adeguata connessione telematica.
- Telelavoro Mobile o Working Out: il lavoratore non ha una sede fissa di lavoro, ma svolge la sua attività spostandosi da un luogo all'altro e comunicando con la sede per mezzo di apparecchiature portatili.
- Centro di Telelavoro o Lavoro Remotizzato: Sono strutture attrezzate con prodotti e servizi tecnologici adatti al telelavoro. In queste strutture si recano i telelavoratori dipendenti o autonomi per fornire le loro prestazioni all'azienda o al committente per cui lavorano. Il centro di telelavoro è una postazione remota rispetto alla sede dell'azienda o del cliente, fornita di dispositivi in grado di consentire la trasmissione e la ricezione di dati;
- Telelavoro nell'impresa virtuale: Molte aziende e gruppi aziendali sono organizzate sulla base di stabilimenti, uffici e centri di produzione dislocati lontano gli uni dagli altri. Il collegamento telematico tra le varie sedi aziendali, i vari reparti, i telelavoratori mobili e domiciliari rende possibile la costruzione di una grande azienda virtuale.

Le tecnologie su cui si basa il telelavoro sono reti pubbliche a banda larga, reti senza fili (wifi, wimax) , reti private virtuali (VPN), protocolli di cifratura, servizi di streaming per videoconferenza.

Il telelavoro, pur essendo riconosciuto in Italia a livello normativo, è scarsamente diffuso. Le limitazioni sono dovute ad aspetti tecnici come la limitata diffusione della banda larga e organizzativi perchè il telelavoro richiede una profonda riorganizzazione delle strutture aziendali.

Analisi

Prima di individuare l'ambiente operativo si effettua l'analisi del problema che prescinde dalla sua implementazione.

Per l'analisi della realtà di interesse si utilizza la modellazione Entità-Relazioni che produce un diagramma ER.

Individuazione delle entità.

Entità 'iscritto': è l'anagrafica degli utenti che frequentano la palestra. Ogni utente deve essere registrato al primo accesso alla palestra in una entità di tipo anagrafico. La registrazione è permanente, indipendentemente dalla frequenza a corsi. L'esigenza di identificare in modo univoco ogni studente porta alla definizione delle seguenti proprietà indispensabili: cognome, nome, sesso, data di nascita, indirizzo, telefono. Trattandosi di un insieme ristretto di persone non è necessaria una completa definizione del tipo 'codice fiscale' e quindi questo insieme non costituisce una chiave candidata. È necessario quindi introdurre una chiave primaria artificiale ad autoincremento.

Una ulteriore informazione indispensabile riguarda il certificato medico. Si suppone che sia possibile iscriversi (ma non partecipare ai corsi) anche senza tale certificato; ma per iscriversi ad un corso è necessario che il certificato sia disponibile e non scaduto (data di emissione più recente di un anno). Si prevedono quindi un campo di tipo data che indica la data di emissione del certificato se consegnato ed è nulla in caso contrario.

I campi sono tutti obbligatori eccetto la data di rilascio del certificato.

Entità tipo di corso': è il catalogo dei corsi che la palestra è in grado di organizzare. Poiché la palestra organizza più edizioni di uno stesso corso questa entità è associata ad una entità debole che definisce i corsi effettivamente organizzati ai quali gli utenti si iscrivono.

Ogni istanza è identificata da una chiave primaria artificiale ad autoincremento e da una descrizione.

Si ipotizza che il numero massimo di iscritti sia sempre lo stesso per ogni tipo di corso quindi è un attributo di questa entità.

Si ipotizza di inserire anche un attributo prezzo sebbene non richiesto esplicitamente dal problema perché il punto di sviluppo dell'interfaccia richiede la presentazione delle 'caratteristiche' dei corsi e il prezzo è sicuramente tra queste.

I campi sono tutti obbligatori.

Entità 'istruttore': è l'anagrafica degli istruttori che dirigono i corsi della palestra. L'esigenza di identificare in modo univoco ogni studente porta alla definizione delle seguenti proprietà indispensabili: cognome, nome. Trattandosi di un insieme ristretto di persone non è necessaria una completa definizione del tipo 'codice fiscale' e quindi questo insieme non costituisce una chiave candidata. È necessario quindi introdurre una chiave primaria artificiale ad autoincremento.

I campi sono tutti obbligatori.

Individuazione delle associazioni.

Entità debole 'corso': Ogni tipo di corso può essere attivato in diverse edizioni che si differenziano per le date/ore di svolgimento.

Ogni utente si iscrive ad un corso effettivamente attivato quindi contraddistinto dall'informazione data/ora.

Questa entità è identificata da una chiave primaria artificiale ad autoincremento e dall'associazione alla tipologia di corso (entità tipo di corso), all'istruttore che lo dirige (si ipotizza che ci sia un solo istruttore per ogni corso attivato) e agli utenti che vi si sono iscritti.

Si ipotizza che ogni corso si svolga solo un giorno alla settimana sempre alla stessa ora quindi si individua un attributo composto giorno/ora. Una soluzione più completa richiederebbe una associazione N:N con i giorni della settimana dotata dell'attributo ora. Ulteriori attributi per storicizzare il corso potrebbero essere la data di inizio e la data di fine corso ma non vengono inseriti perché non richiesti dal problema.

Associazione iscritto/corso 'iscrizione': ogni iscritto può frequentare nessuno, uno o più corsi attivati e un corso attivato può essere composto da nessuno, uno o più utenti iscritti. Esiste quindi una associazione N:N tra iscritto e corso. La relazione è parziale da entrambi i lati (un utente può non essere iscritto ad alcun corso e un corso può non avere alcun iscritto). Questa associazione potrebbe avere un attributo booleano 'pagato' per la gestione economica ma poiché non è richiesta dal problema non viene implementata.

Associazione corso/tipo di corso 'appartiene': ogni corso effettivamente attuato deve essere associata ad un tipo di corso. Esiste quindi una associazione 1:N tra tipo di corso e corso. La relazione è parziale dal lato tipo di corso e totale dal lato corso (un tipo di corso può non essere associato ad alcuno corso, un corso deve essere associato ad un tipo di corso.)

Associazione istruttore/corso 'diretto_da': ogni corso effettivamente attuato deve essere diretto da un singolo istruttore; un istruttore può dirigere più corsi. Esiste quindi una associazione 1:N tra istruttore e corso. La relazione è parziale dal lato istruttore e totale dal lato corso (un tipo di corso può non essere associato ad alcuno corso, un corso deve essere associato ad un tipo di corso.)

Ipotesi aggiuntive

Si riassumono qui le principali ipotesi aggiuntive già presentate nella analisi di entità ed associazioni.

Anagrafe iscritti: in considerazione del limitato numero di istanze di questa anagrafe non si inseriscono i dati completi di identificazione anagrafica (tipo codice fiscale) né di residenza ma solo un insieme ridotto essenziale per l'identificazione dell'utente. In particolare, non si inserisce il comune di nascita, non si effettua la decomposizione dell'attributo composto indirizzo né la normalizzazione del comune di residenza.

Certificato medico: la partecipazione ai corsi è indipendente dall'iscrizione alla palestra quindi l'iscrizione può essere fatta anche in assenza di certificato medico; per partecipare ad un corso è necessario avere presentato un certificato medico non scaduto. L'attributo data_rilascio dell'entità iscritto è facoltativo. Se è nullo vuole dire che il certificato non è stato ancora consegnato altrimenti indica la data di rilascio. Un'alternativa alla nullità potrebbe essere l'impostazione del default a '00-00-0000' che determina un certificato sicuramente scaduto.

Numero massimo di iscritti: si ipotizza che il numero massimo di iscritti sia uguale per tutti i corsi dello stesso tipo quindi il numero massimo è attributo del tipo di corso. Una soluzione alternativa che consenta di cambiare il numero massimo di iscritti per ogni corso effettuato richiederebbe di trasportare questo attributo nell'entità debole 'corso'.

Prezzo di un corso: sebbene non sia esplicitamente richiesto viene inserito l'attributo 'prezzo' nell'entità corso. Il punto di sviluppo dell'interfaccia richiede in generale la presentazione delle caratteristiche dei corsi e il prezzo è sicuramente una informazione importante per l'utente. Si ipotizza che il prezzo sia uguale per tutti i corsi dello stesso tipo quindi il prezzo è attributo del tipo di corso. Una soluzione alternativa che consenta di cambiare il prezzo per ogni corso effettuato richiederebbe di trasportare questo attributo nell'entità debole 'corso'.

Anagrafe istruttori: in considerazione del limitato numero di istanze di questa anagrafe e del fatto che non sono richieste transazioni di tipo amministrativo (stipendio, previdenza ...) non si inseriscono i dati completi di identificazione anagrafica (tipo codice fiscale) né di residenza ma solo un insieme ridotto essenziale per l'identificazione dell'utente. In particolare si inserisce solo il nome e cognome.

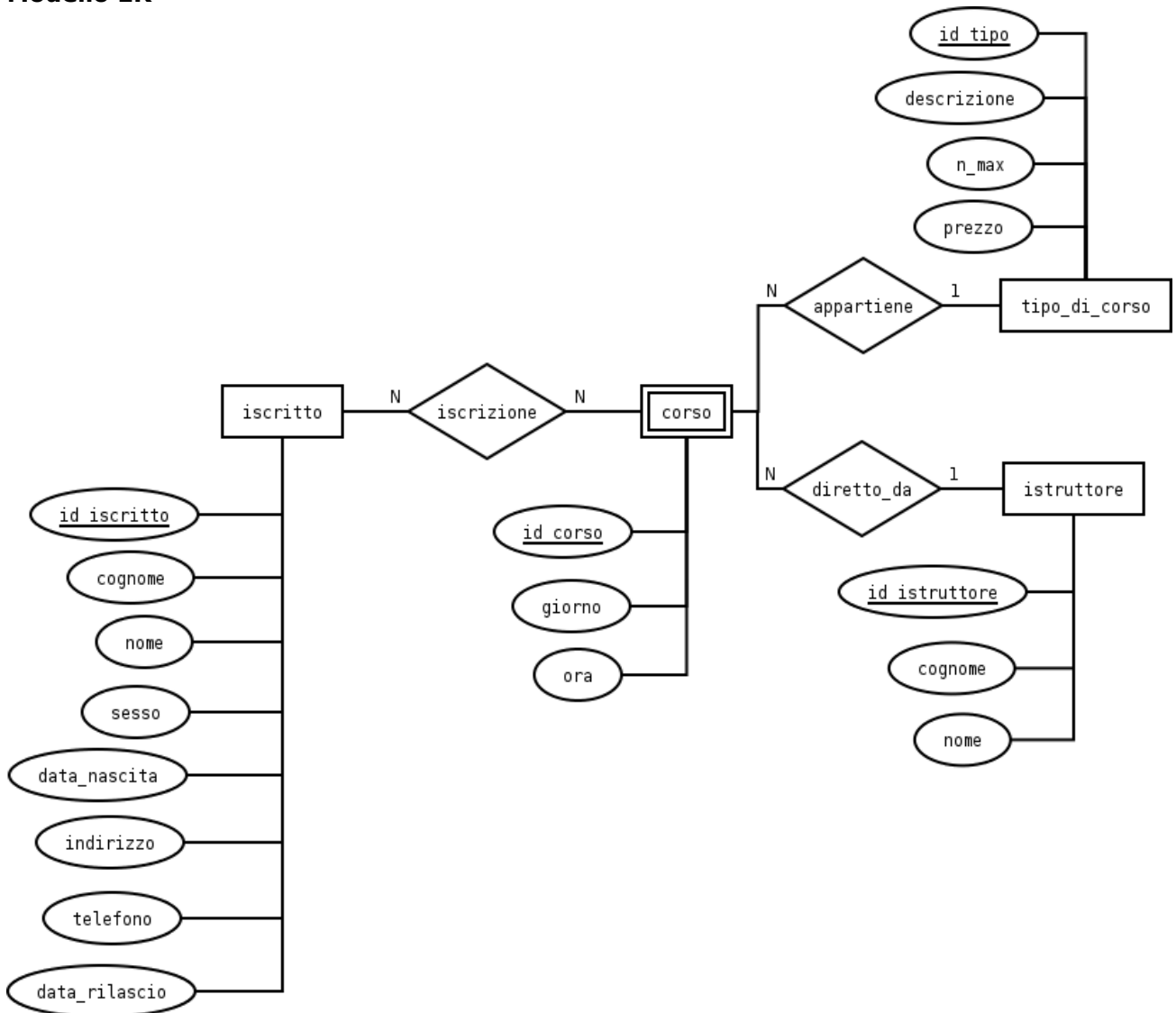
Periodo di attuazione di un corso: si ipotizza che un corso attuato si svolga solo una volta alla settimana sempre nella stessa ora. Si inserisce quindi un attributo composto giorno/ora che identifica il periodo di svolgimento del corso.

Una soluzione alternativa più complessa avrebbe richiesto una associazione N:N tra 'corso' e l'entità di look-up 'giorni_della_settimana' dotata dell'attributo 'ora' consentendo di distribuire un corso su più giorni della settimana anche in ore diverse (una sola lezione ogni giorno).

Una ulteriore informazione che non viene messa perché non richiesta è la data inizio e data fine che consentirebbe di storicizzare il corso mantenendo traccia delle attività svolte.

Direzione dei corsi: si ipotizza che un corso sia diretto da un solo istruttore. Una soluzione alternativa di conduzione multipla avrebbe richiesto una associazione N:N tra istruttore e corso.

Modello ER



Schema logico

Si usa il modello relazionale.

Ogni entità ed entità debole viene sostituita con una relazione.

Ogni associazione 1:N tra due entità viene sostituita con l'esportazione della chiave primaria del lato 1 come chiave esterna nel lato N.

Ogni associazione N:N tra due entità viene sostituita con la generazione di una nuova relazione ottenuta esportando le due chiavi primarie come chiavi esterne e definendo la loro composizione come chiave primaria della nuova relazione.

```
iscritto(id_iscritto:intero,
        cognome:testo,
        nome:testo,
        sesso:logico,
        data_nascita: data,
        indirizzo: testo,
        telefono: testo,
        data_rilascio: data)
```

```
tipo_di_corso(id_tipo:intero,
              descrizione: testo,
              nmax:intero,
              prezzo: valuta)
```

```
istruttore(id_istruttore,
            cognome: testo ,
            nome: testo)
```

```
corso(id_corso. intero,
      giorno: testo,
      ora: testo,
      id_tipo: intero,
      id_istruttore: intero)
```

```
iscrizione(id_iscritto: intero, id_corso: intero)
```

Vincoli di integrità

Vincoli di integrità referenziale

```
corso.id_tipo ⊆ tipo_di_corso.id_tipo
corso.id_istruttore ⊆ istruttore.id_istruttore
iscrizione.id_iscritto ⊆ iscritto.id_iscritto
iscrizione.id_corso ⊆ corso.id_corso
```

Vincoli esterni

```
tipo_di_corso.n_max > 0
tipo_di_corso.prezzo > 0
corso.giorno ∈ {'lunedì'; 'martedì'; 'mercoledì'; 'giovedì'; 'venerdì'; 'sabato'}
corso.ora ∈ {'8:00'; '9:00'; ...; '21:00'}
iscritto.sesso ∈ {'M'; 'F'}
se certificato non consegnato data_rilascio = NULL
```

Modello fisico

Sebbene non sia esplicitamente richiesto si sviluppa il modello fisico con la sintassi SQL per mettere in evidenza la realizzazione e scelte fatte sui tipi di dati e sui vincoli.

```
CREATE TABLE iscritto (  
  id_iscritto int(11) NOT NULL auto_increment,  
  cognome varchar(20) NOT NULL default '',  
  nome varchar(20) NOT NULL default '',  
  sesso set('M','F') NOT NULL default 'M',  
  data_nascita date NOT NULL default '0000-00-00',  
  indirizzo varchar(80) NOT NULL default '',  
  telefono varchar(20) NOT NULL default '',  
  data_rilascio date default NULL,  
  PRIMARY KEY (id_iscritto)  
);  
  
CREATE TABLE istruttore (  
  id_istruttore int(11) NOT NULL auto_increment,  
  cognome varchar(20) NOT NULL default '',  
  nome varchar(20) NOT NULL default '',  
  PRIMARY KEY (id_istruttore)  
);  
  
CREATE TABLE tipo_di_corso (  
  d_tipo int(11) NOT NULL auto_increment,  
  descrizione varchar(20) NOT NULL default '',  
  n_max int(10) unsigned NOT NULL default '0',  
  prezzo decimal(8,2) NOT NULL default '0.00',  
  PRIMARY KEY (d_tipo)  
);  
  
CREATE TABLE corso (  
  id_corso int(11) NOT NULL auto_increment,  
  giorno set('lunedì','martedì','mercoledì','giovedì','venerdì','sabato') NOT NULL,  
  ora set('9:00','10:00','11:00','12:00','21:00') NOT NULL default '9:00',  
  id_tipo int(11) NOT NULL,  
  id_istruttore int(11) NOT NULL,  
  PRIMARY KEY (id_corso),  
  INDEX idx_id_tipo(id_tipo),  
  INDEX idx_id_istruttore(id_istruttore),  
  FOREIGN KEY(id_tipo) REFERENCES tipo_di_corso(id_tipo),  
  FOREIGN KEY(id_istruttore) REFERENCES istruttore(id_istruttore)  
);  
  
CREATE TABLE iscrizioni (  
  id_iscritto int(11) NOT NULL default '0',  
  id_corso int(11) NOT NULL default '0',  
  PRIMARY KEY (id_iscritto,id_corso),  
  INDEX idx_id_iscritto(id_iscritto),  
  INDEX idx_id_corso(id_corso),  
  FOREIGN KEY(id_iscritto) REFERENCES iscritto(id_iscritto),  
  FOREIGN KEY(id_corso) REFERENCES corso(id_corso)  
);
```

Scelta dell'ambiente operativo

Si tratta di una piccola realtà locale con un numero limitato di istanze. Si può quindi ipotizzare la scelta di un ambiente operativo di tipo Personal Data Base come l'ambiente Base di OpenOffice oppure l'ambiente Access di Microsoft.

Entrambi questi ambienti costituiscono applicazioni integrate che contengono in un unico documento tutte le risorse necessarie per la realizzazione del database:

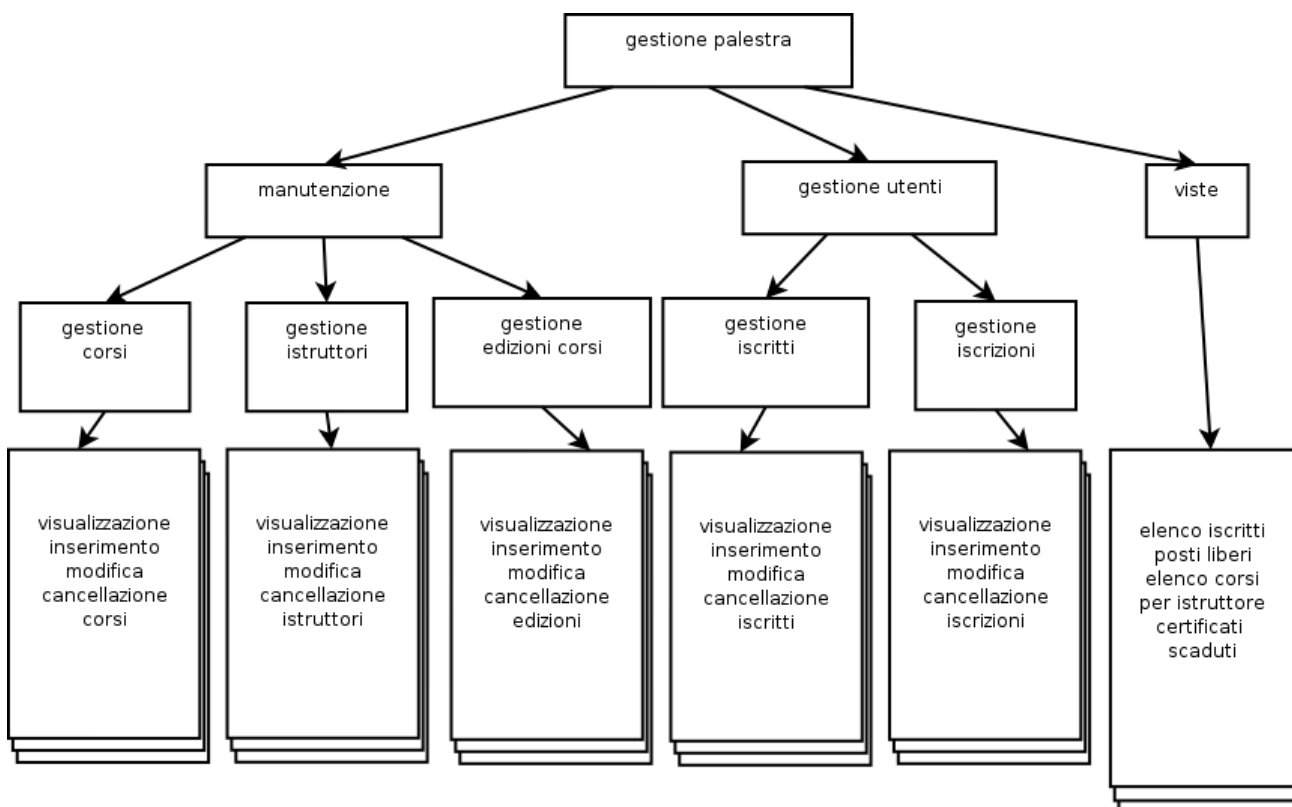
- Tabelle
- Query
- Maschere
- Report
- Macro

L'ambiente applicativo ed il relativo motore di DBMS sono installati su una workstation locale e sono accessibili solo da tale workstation o, in presenza di una rete locale, dalle altre workstation della LAN mediante una condivisione.

Progettazione dell'interfaccia

La richiesta di un menu di scelta porta all'ipotesi di realizzare un 'pannello comandi' inserito nella macro di autoavvio che proponga le possibili operazioni sul database.

Viene sviluppato l'albero dei collegamenti tra le maschere:



Poiché la parte finale della traccia richiede la pubblicazione web di almeno una parte delle informazioni della banca dati si può fare una ipotesi alternativa interamente basata su applicazioni web.

Si può ipotizzare ad esempio di usare una piattaforma xAMP composta da:

- x sistema operativo Windows o Linux
- A web server Apache
- M sql server MySQL
- P script interprete PHP

Interrogazioni

elenco degli iscritti a ogni corso

Si effettua una giunzione naturale tra le entità iscritto e corso legate dalla associazione iscrizione per trovare gli iscritti ai corsi ed i dettagli del corso e si aggiunge una giunzione naturale a tipo di corso per estrarre la descrizione del corso.

Non viene effettuata alcuna restrizione ad eccetto di quelle relative al join implicito

La proiezione fornisce un elenco contenente i dati del corso e i dati dell'iscritto ordinati per corso ed iscritto.

```
SELECT
    t4.descrizione as descorso,
    t3.giorno,
    t3.ora,
    t1.cognome,
    t1.nome
FROM
    iscritto AS t1,
    iscrizione AS t2,
    corso AS t3,
    tipo_di_corso AS t4
WHERE
    t1.id_iscritto=t2.id_iscritto
AND
    t2.id_corso=t3.id_corso
AND
    t3.id_tipo=t4.id_tipo
ORDER BY
    t3.id_corso, t1.cognome, t1.nome
```

numero dei posti liberi in ciascun corso

La query è simile alla precedente ma non è presente nella giunzione l'entità iscritto perché non interessano i dati dei singoli iscritti. La giunzione naturale è quindi tra l'associazione iscrizione e il solo lato corso per ottenere gli id di iscrizione e i dettagli del corso, in giunzione naturale con tipo di corso per estrarre la descrizione e il numero massimo.

Viene fatta una aggregazione in base all'id del corso attuato.

La proiezione effettua una differenza tra il numero massimo di iscritti ed il conteggio degli iscritti (id_iscritto) di ciascun corso.

La restrizione sul raggruppamento (HAVING) estrae i corsi che hanno posti liberi (conteggio maggiore di n_max)

```
SELECT
    t4.descrizione as descorso,
    t3.giorno,
    t3.ora,
    t4.n_max - count(t2.id_iscritto) as posti_liberi
FROM
    iscrizioni AS t2,
    corso AS t3,
    tipo_di_corso AS t4
WHERE
    t2.id_corso=t3.id_corso
AND
    t3.id_tipo=t4.d_tipo
GROUP BY
    t2.id_corso
HAVING
    posti_liberi > 0
ORDER BY
    t3.id_corso
```

elenco dei corsi tenuti da ogni istruttore

Si effettua una giunzione naturale tra le entità corso, tipo di corso ed istruttore per estrarre rispettivamente dati del corso attuato e direzione del corso, descrizione del corso e dati anagrafici dell'istruttore che lo dirige.

Non viene effettuata alcuna restrizione ad eccetto di quelle relative al join implicito

La proiezione fornisce un elenco contenente i dati dell'istruttore e dei suoi corsi corso ordinati per istruttore e corso.

```
SELECT
    t3.cognome,
    t3.nome,
    t2.descrizione,
    t1.giorno,
    t1.ora
FROM
    corso AS t1,
    tipo_di_corso AS t2,
    istruttore as t3
WHERE
    t1.id_tipo=t2.id_tipo
AND
    t1.id_istruttore=t3.id_istruttore
ORDER BY
    t3.cognome, t3.nome, t2.descrizione,t1.giorno,t1.ora
```

elenco dei clienti con certificato medico scaduto

L'estrazione delle informazioni avviene nell'unica tabella iscritto con la restrizione sulla data di rilascio del certificato.

Si usa la funzione datediff() applicando la differenza tra la data corrente e la data di rilascio del certificato.

Da questa restrizione sfuggono gli iscritti che non hanno portato il certificato e quindi hanno NULL nel campo data_rilascio.

Vengono aggiunti testando in OR la nullità di tale campo.

Nella proiezione viene mostrato il nome cognome e data di rilascio del certificato scaduto (NULL se non consegnato)

```
SELECT
    t1.cognome,
    t1.nome,
    t1.data_rilascio)
FROM
    iscritto AS t1
WHERE
    DATEDIFF(CURDATE(),t1.data_rilascio)
OR
    ISNULL(t1.data_rilascio)
ORDER BY
    t1.cognome,t1.nome
```

Interfaccia web

Si ipotizza di usare una piattaforma xAMP composta da:

- x sistema operativo Windows o Linux
- A web server Apache
- M sql server MySQL
- P script interprete PHP

pagina web con la quale la palestra presenta alla clientela i corsi, le relative caratteristiche e gli orari.

La pagina effettua una connessione alla banca dati e termina immediatamente se la connessione fallisce.

Se la connessione ha successo viene richiesta una query che effettua un join tra iscrizione e corsi per estrarre i dati di ogni corso e i suoi iscritti che saranno usati nel conteggio; un ulteriore join con tipo di corso estrae il numero massimo di posti ed il prezzo mentre un join con istruttore estrae il nome/cognome dell'istruttore.

Il risultato della giunzione viene aggregato per corso e il numero di iscritti viene contato per ogni corso; in questo modo è possibile presentare il numero di posti disponibili in ogni corso facendo la differenza con il numero di posti massimo.

```

<?php
$sock=mysql_connect('localhost','nobody','qwerty');
if ($sock==0) die(mysql_error());
$ris=mysql_select_db('palestra',$sock);
if ($ris==0) die(mysql_error());
$msg="SELECT
        t3.descrizione,
        t2.giorno,
        t2.ora,
        t3.n_max,
        t3.n_max - COUNT(t1.id_iscritto) as posti_liberi,
        t2.prezzo,
        CONCAT(t3.nome,t3.cognome) AS istr
FROM
        iscrizione AS t1
        corso as t2,
        tipo_di_corso as t3,
        istruttore as t4
WHERE t1.id_iscritto=t2.id_iscritto
AND   t2.id_tipo=t3.id_tipo
AND   t2.id_istruttore=t4.id_istruttore
GROUP BY
        t2.id_corso
ORDER BY
        t3.descrizione,t2.giorno,t2.ora";
$query=mysql_query($msg,$sock);
if ($query==0) die(mysql_error());
?>
<html>
<head>
...
</head>
<body>
<table>
<tr>
<td>corso</td>
<td>giorno</td>
<td>ora</td>
<td>prezzo</td>
<td>posti max</td>
<td>posti disponibili</td>
<td>istruttore</td>
</tr>
<?php while($row_corsi=mysql_fetch_assoc($query)) { ?>
<tr>
<td><?php echo $row_corsi['descrizione'] ?></td>
<td><?php echo $row_corsi['giorno'] ?></td>
<td><?php echo $row_corsi['ora'] ?></td>
<td><?php echo $row_corsi['n_max'] ?></td>
<td><?php echo $row_corsi['prezzo'] ?></td>
<td><?php echo $row_corsi['posti_liberi'] ?></td>
<td><?php echo $row_corsi['istr'] ?></td>
</tr>
<?php } ?>
</table>
</body>
</html>

```