

Esame di Stato Istituto Tecnico Commerciale
Proposta di soluzione della seconda prova
Indirizzo: PROGRAMMATORI - CORSO DI ORDINAMENTO
Tema di: INFORMATICA GENERALE ED APPLICAZIONI GESTIONALE
Anno Scolastico: 2004-2005

Introduzione

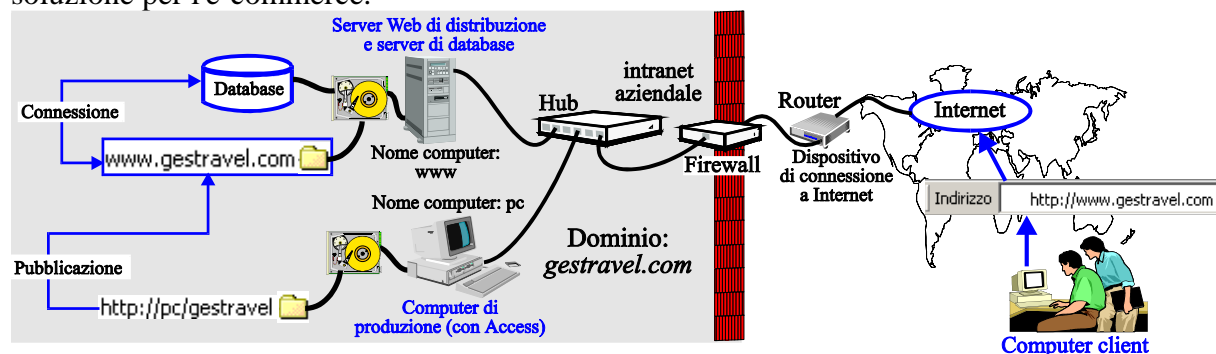
Il telelavoro riguarda un lavoratore dipendente che svolge le sue mansioni al di fuori dell'azienda, spesso nella sua abitazione.

E' una forma di lavoro flessibile, che implica un cambiamento nella struttura organizzativa dell'impresa. Per svolgere il telelavoro è necessario un computer che sia collegato, in modo remoto via Internet, ai computer server dell'impresa di cui il telelavoratore è dipendente. Le connessioni remote che permettono l'accesso ai server dell'azienda ai lavoratori esterni devono essere protette, ad esempio, mediante la creazione di connessioni cifrate.

Per e-commerce si intende l'insieme delle contrattazioni riguardanti le cessioni di beni e le prestazioni di servizi che avvengono in via telematica.

I motivi per cui il telelavoro e l'e-commerce non sono particolarmente sviluppati in Italia riguardano principalmente aspetti culturali e organizzativi.

La figura seguente presenta un esempio delle strutture informatiche necessarie a un'azienda per offrire l'accesso via Internet ai propri sistemi informativi interni e quindi realizzare una soluzione per l'e-commerce.



Analisi e progetto del sistema informativo

Lo schema concettuale (o statico) è uno dei due schemi del progetto concettuale di un sistema informativo.

Fatti elementari

Partendo dalle specifiche, contenute nel testo del problema, i fatti elementari del sistema informativo in fase di progetto sono i seguenti.

- | |
|---|
| ■ Un istruttore può insegnare in più corsi, ma un corso può avere più istruttori. |
| ■ Un corso può essere frequentato da più clienti. |
| ■ Ad ogni corso sono associati più orari (in giorni e/o ore diverse). |

Tipi di entità e attributi

I tipi di entità caratteristici del sistema sono:

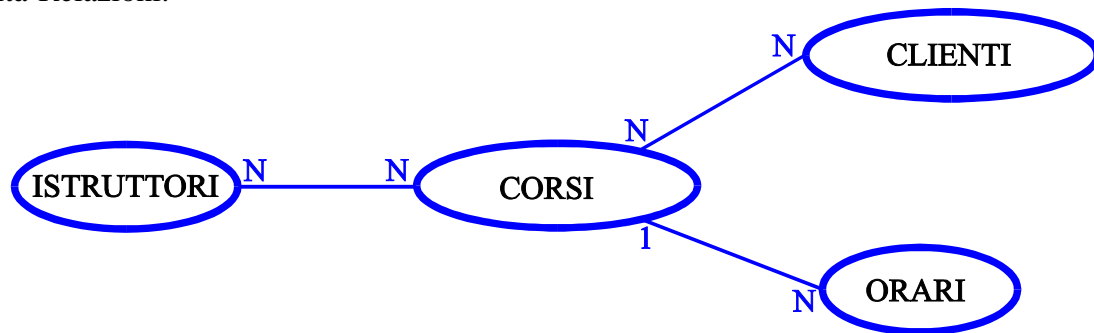
Corsi
Istruttori
Clienti
Orari

I tipi di entità vengono individuati dai gruppi di attributi illustrati in modo dettagliato nella figura che segue.



Le relazioni

Le relazioni tra i tipi di entità nel modello dei dati sono individuate nel seguente diagramma Entità-Relazioni.



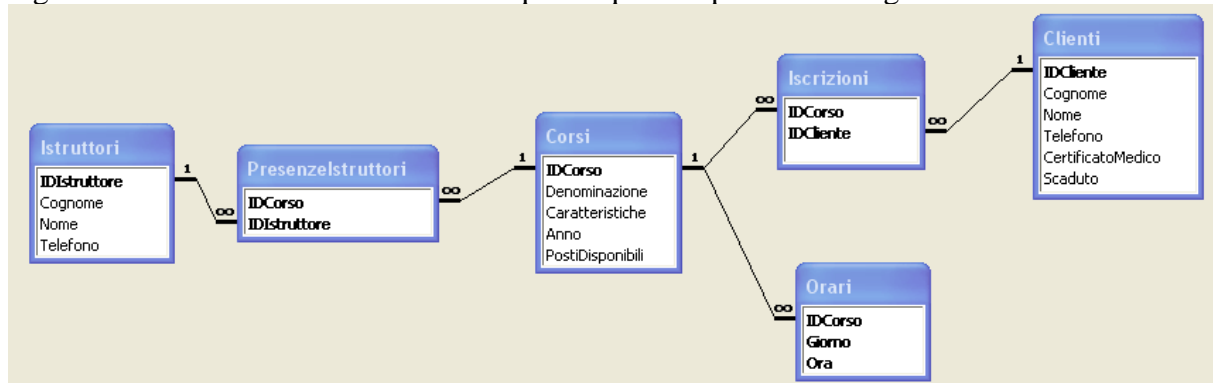
Schema logico della base di dati

Passaggio dallo schema statico del progetto concettuale al modello relazionale

Per la creazione dello schema logico della base di dati relazionale effettueremo i seguenti due passi.

1. Traduzione dello schema statico concettuale in un modello relazionale preliminare.
2. Applicazione delle tre forme normali per verificare se lo schema preliminare può essere ulteriormente ottimizzato.

Nella fase di realizzazione del progetto concettuale, le relazioni molti a molti *Istruttori-Corsi* e *Corsi-Clienti* devono essere trasformate in due associazioni uno a molti e molti a uno, introducendo, rispettivamente, le tabelle ausiliarie *PresenzeIstruttori* e *Iscrizioni*. Lo schema logico relazionale del database diventa quindi quello riportato nel seguito.



L'applicativo da utilizzare deve essere un RDBMS (*Relational-Database Management Sistema*). Nella soluzione proposta ci riferiremo al RDBMS Access (realizzato con il file di database *Palestra.mdb*).

Anche se non richiesto, nel seguito realizzeremo tutte le operazioni del testo della prova d'Esame.

Implementazione delle operazioni in linguaggio SQL

Operazione 1: elenco degli iscritti a ogni corso.

Codice sorgente SQL

```
SELECT Clienti.Cognome, Clienti.Nome
FROM Clienti, Iscrizioni, Corsi
WHERE Clienti.IDcliente = Iscrizioni.IDcliente AND
      Iscrizioni.IDCorso = Corsi.IDCorso AND
      Corsi.IDCorso = [Introduci il codice del corso:]
```

Nel codice precedente, così come in quelli dei servizi successivi, i nomi racchiusi tra parentesi quadre (ad esempio *[Introduci il codice del corso:]*) rappresentano parametri di ingresso il cui valore viene richiesto dal RDBMS prima dell'esecuzione dell'operazione.

Operazione 2: il numero di posti liberi in ciascun corso.

Codice sorgente SQL

```
SELECT Corsi.PostiDisponibili -
      (SELECT COUNT(Iscrizioni.IDcliente)
       FROM Iscrizioni INNER JOIN Corsi ON
           Iscrizioni.IDCorso = Corsi.IDCorso
       WHERE Corsi.IDCorso = [Introduci il codice del corso:])
AS 'Numero posti liberi'
FROM Corsi
WHERE Corsi.IDCorso = [Introduci il codice del corso:]
```

Osserviamo che l'operazione precedente effettua la differenza tra il numero di posti disponibili in ogni corso e il numero di posti in un corso dato dalla seguente query (inserita come subquery).

```
SELECT COUNT(Iscrizioni.IDcliente)
FROM Iscrizioni INNER JOIN Corsi ON Iscrizioni.IDCorso = Corsi.IDCorso
```

Operazione 3: elenco dei corsi tenuti da ogni istruttore.

Codice sorgente SQL

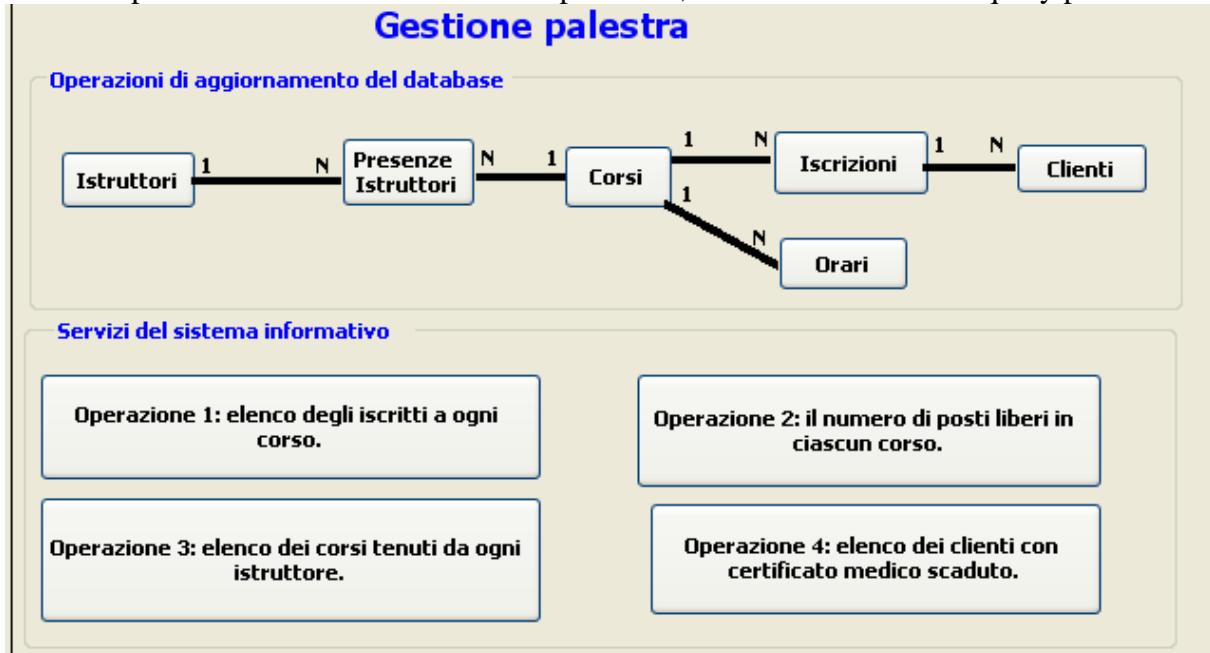
```
SELECT Corsi.Denominazione
FROM Istruttori, PresenzeIstruttori, Corsi
WHERE Istruttori.IDIstruttore = PresenzeIstruttori.IDIstruttore AND
      PresenzeIstruttori.IDCorso = Corsi.IDCorso AND
      Istruttori.Cognome = [Cognome istruttore:] AND
      Istruttori.Nome = [Nome istruttore:]
```

Operazione 4: elenco dei clienti con certificato medico scaduto.

Codice sorgente SQL

```
SELECT Clienti.Cognome, Clienti.Nome
FROM Clienti
WHERE Scaduto = Yes
```

Il menu di scelta può anche essere realizzato con una maschera del RDBMS Access che permette di aprire le tabelle del database e le operazioni, realizzate mediante le query precedenti.



Creazione della pagina Web

Il sito può essere organizzato con una home page (*default.htm*) a frame che permette di richiamare le pagine con le operazioni richieste.



Le operazioni possono essere realizzate con una pagina HTML con un modulo (solo se è necessario l'input dei parametri dell'interrogazione) che richiama una pagina ASP che effettua la query richiesta sul database *Palestra.mdb*.

La **connessione al database** può essere dichiarata come variabile globale in modo che sia visibile durante una sessione HTTP di un utente all'applicazione Web. Per questo obiettivo è sufficiente creare un file di testo di nome *global.asa*, nella radice del sito Web, e dichiarare la variabile *StringaConnessione* nel corpo dell'evento *Session_OnStart* con la seguente sintassi.

```

global.asa
<SCRIPT LANGUAGE="VBScript" RUNAT="SERVER">
Sub Session_OnStart()
  Dim StringaConnessione
  Session("StringaConnessione")="DRIVER={Microsoft Access Driver (*.mdb)};" &
    "DBQ="& Server.MapPath("../fpdb/Palestra.mdb")
End Sub
</SCRIPT>
  
```

Operazione: elenco dei corsi disponibili.

Per realizzare questa operazione, possiamo progettare la pagina ASP *elenco_corsi.asp*, richiamata nel menu del sito Web con l'hyperlink [Corsi](#). Lo script (riportato nel seguito) inserito nella pagina ASP esegue la query per la ricerca di tutti i corsi della palestra memorizzati nel database e presenta una tabella con i risultati visualizzati in una pagina HTML dinamica di risposta.

```
corsi/elenco_corsi.asp
<%@ Language=VBScript %>
<% Response.Buffer=True %>

<HTML>
<HEAD><TITLE>Elenco dei corsi della palestra </TITLE></HEAD>
<BODY>
<H3>Caratteristiche del corso</H3>
<% Dim ConnessioneDB, rsTabella, Query
   Set ConnessioneDB = Server.CreateObject("ADODB.Connection")
   ConnessioneDB.Open Session("StringaConnessione")

   Query="SELECT * " &_
        " FROM Corsi "

   Set rsTabella = ConnessioneDB.Execute(Query)%>
<% If rsTabella.BOF=True And rsTabella.EOF=True Then %>
  <H4>Nessun corso corrisponde al codice digitato </H4>
<% Else %>
  <TABLE BORDER="1">
  <TR><TH>Codice</TH><TH>Denominazione</TH><TH>Caratteristiche</TH><TH>Anno</TH>
  </TR>
  <% Do until rsTabella.EOF=True %>
    <TR>
      <TD ALIGN="CENTER"><%=rsTabella.Fields("IDCorso")%></TD>
      <TD ALIGN="CENTER"><%=rsTabella.Fields("Denominazione")%></TD>
      <TD ALIGN="CENTER"><%=rsTabella.Fields("Caratteristiche")%></TD>
      <TD ALIGN="CENTER"><%=rsTabella.Fields("Anno")%></TD>
    </TR>
    <% rsTabella.MoveNext %>
  <% Loop %>
  </TABLE>
<% End If %>

<% rsTabella.Close : Set rsTabella = Nothing
   ConnessioneDB.Close : Set ConnessioneDB = Nothing
   Session.Abandon %>
</BODY>
</HTML>
```