

Esame di Stato Istituto Tecnico Commerciale
CORSO SPERIMENTALE – Progetto "MERCURIO"
Indirizzo: PROGRAMMATORI
Soluzione della Seconda Prova - Tema di: INFORMATICA
Anno Scolastico: 2002-2003

Riflessioni e apparecchiature necessarie

L'aumento del numero di computer e degli utenti connessi a Internet causa due sostanziali problematiche:

- l'aumento degli indirizzi IP e dei corrispondenti gruppi di nomi di dominio assegnati;
- l'aumento del traffico nelle backbone (dorsali) della Rete.

Il primo problema è stato parzialmente risolto con l'assegnazione di gruppi limitati di indirizzi IP pubblici (con subnet mask binarie), invece di assegnare gruppi di 256 indirizzi alla volta (come con le subnet mask decimali). Un'ulteriore soluzione, in fase di studio e sperimentazione nel protocollo *IPv6*, prevede l'aumento del numero di bit degli indirizzi IP da 32 a 64, con il conseguente aumento degli indirizzi disponibili a 2^{64} . L'aumento del traffico in Internet deve essere risolto a livello dei sottosistemi di telecomunicazione, che devono garantire velocità di trasmissione (in bps) e bande di frequenza sempre maggiori. Esempi di queste nuove infrastrutture sono le linee ADSL e le reti ATM, quest'ultime impiegate nella costruzione dei nodi delle reti a commutazione di pacchetto sia di Internet sia delle reti telefoniche digitali.

I principali servizi di Internet utili per le soluzioni per l'e-commerce sono: il World Wide Web (WWW) o semplicemente Web, la posta elettronica, il trasferimento dei file (FTP) e l'accesso ai database.

Per rendere visibile un computer in Internet è necessario disporre di uno oppure più indirizzi IP pubblici, ai quali è associato un nome di dominio di secondo livello del tipo, ad esempio, *azienda.com*. Ogni indirizzo IP pubblico è quindi associato a un computer server che offre uno oppure più servizi. Ad esempio, il computer *www.azienda.com* può offrire i servizi WWW e FTP, mentre il server *mail.azienda.com*, può offrire i servizi di posta elettronica. L'accesso ad una oppure più basi di dati può essere garantito installando un server di database nella rete privata dell'azienda, che può anche coincidere con lo stesso server Web.

Le apparecchiature hardware necessarie per offrire servizi on-line per Internet sono:

- server Web e di database, realizzati con computer di classe server di elevate prestazioni, in termini di CPU, RAM e velocità degli hard disk;
- un router, che permetta la connessione di tutta la rete di un'azienda (intranet) alla Rete, con indirizzi IP privati, impiegando un indirizzo IP pubblico;
- una connessione permanente a Internet con una banda garantita e quindi velocità di trasmissione elevate (ad esempio, 1-2 Mbps), realizzata ad esempio con una linea ADSL non commutata;
- un firewall, per garantire la protezione dagli attacchi esterni di persone non autorizzate da Internet (*cracker*) nell'intranet dell'azienda.

Caratteristiche dell'applicativo da utilizzare

Per sviluppare il servizio on-line dell'Istituto sono necessari i programmi per realizzare un server Web e un server di database. Per il servizio WWW, è necessario installare in un computer di classe server un programma di sistema quale, ad esempio, Apache per i server UNIX/Linux oppure IIS per i server Windows 2000.

Per realizzare un server di database, è necessario installare un RDBMS (*Relational DataBase Management System*) in grado di permettere la realizzazione dello schema logico relazionale della base di dati e automatizzare le sue procedure di gestione. Il server di database può essere un computer autonomo della rete aziendale oppure coincidere con il server Web. L'accesso ai dati è realizzato sviluppando alcune pagine da inserire nel sito Web che permettono di eseguire interrogazioni sul server di database e presentare agli utenti i risultati mediante pagine HTML dinamiche di risposta. Le pagine di accesso ai dati contengono degli script server-side (eseguiti dal server) che aprono connessioni al server di database, eseguono query (anche con parametri di ingresso) e presentano i risultati agli utenti (generando pagine HTML dinamiche).

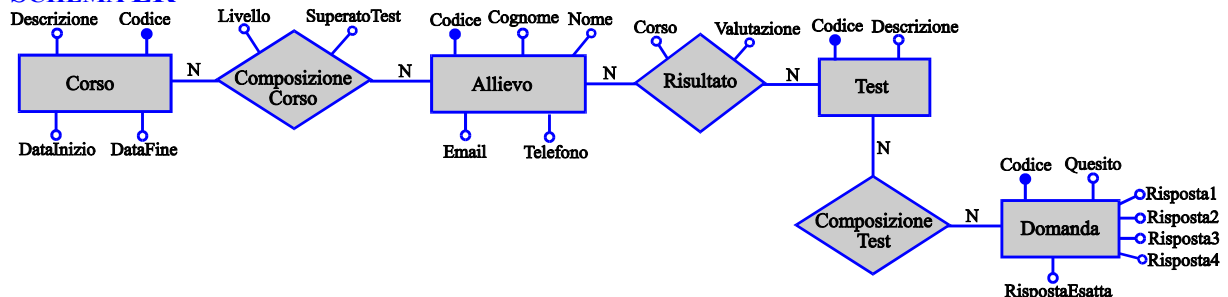
Nel seguito, utilizzeremo Access come RDBMS, mentre per realizzare l'accesso ai dati seguiremo la tecnologia ASP basata sul linguaggio di scripting VBScript.

Schema generale della procedura e organizzazione dei dati

Per la realizzazione on-line dei servizi dell'Istituto possiamo pensare al seguente progetto concettuale e schema logico relazionale del database, che soddisfa le forme normali.

■ ■ ■ PROGETTO CONCETTUALE

SCHEMA ER



Lo schema ER deve seguire queste **regole**.

- R1.** Lo schema precedente permette di gestire qualsiasi tipo di *Test*, che nel nostro caso potrebbero essere codificati con *Codice*="TING-1", per i test di ingresso del *Corso* con *Codice*="CORSO1", e *Codice* del *Test* uguale a "TFIN-1", per il test finale del corso "CORSO1".
- R2.** L'attributo *Valutazione* di un *Risultato* memorizza un numero reale con la percentuale di risposte esatte sul totale con cui un allievo ha effettuato un test d'ingresso per un determinato corso (ad esempio 78,67, rappresentato in percentuale 78,67%).
- R3.** L'attributo *Codice* di una *Domanda* è un numero intero scelto manualmente dall'operatore.
- R4.** L'attributo *RispostaEsatta* di una *Domanda* è un numero intero che rappresenta il numero (da 1 a 4) della risposta corretta.

■ ■ ■ REALIZZAZIONE

SCHEMA LOGICO RELAZIONALE



Lo schema logico relazionale **rispetta le forme normali** ed è soggetto a questi **vincoli di integrità referenziale**.

- V1. La chiave esterna *Corso* della tabella *ComposizioneCorsi* è in relazione con la tabella *Corsi* mediante la chiave primaria *Codice*.
- V2. La chiave esterna *Allievo* della tabella *ComposizioneCorsi* è in relazione con la tabella *Allievi* mediante la chiave primaria *Codice*.
- V3. La chiave esterna *Allievo* della tabella *Risultati* è in relazione con la tabella *Allievi* mediante la chiave primaria *Codice*.
- V4. La chiave esterna *Test* della tabella *Risultati* è in relazione con la tabella *Test* mediante la chiave primaria *Codice*.
- V5. La chiave esterna *Test* della tabella *ComposizioneTest* è in relazione con la tabella *Test* mediante la chiave primaria *Codice*.
- V6. La chiave esterna *Domanda* della tabella *ComposizioneTest* è in relazione con la tabella *Domande* mediante la chiave primaria *Codice*.

CODIFICA

Lo **schema logico del database** può essere realizzato in Access (in alternativa).

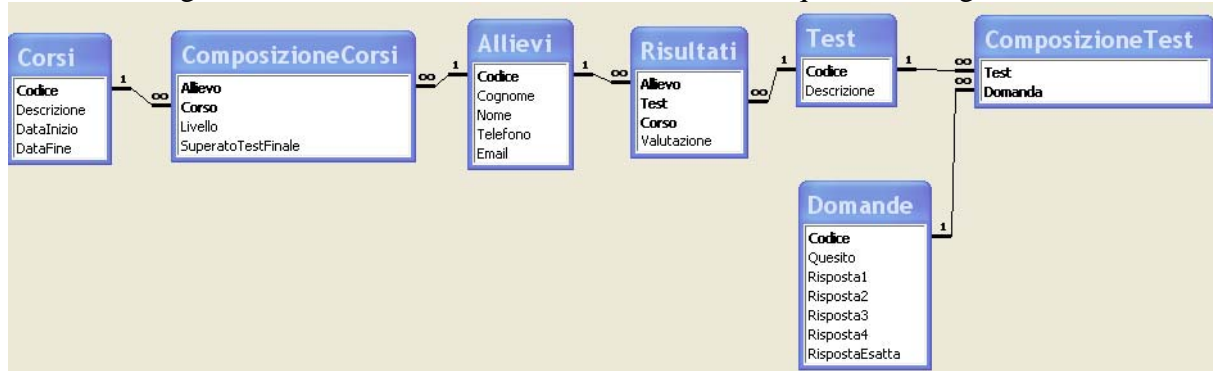
- In modo interattivo, definendo le singole tabelle di base e impostando le relazioni.
- Nella modalità programma, eseguendo nell'ordine i quattro script SQL, documentati nella tabella che segue.

Il database può essere memorizzato nel file *DBCorsiOnLine.mdb*. La sintassi standard SQL è stata adattata a quella implementata in Access.

Tabelle di base	Codice SQL nella modalità SQL Visualizzazione SQL												
<div style="border: 1px solid black; padding: 5px;"> <p>Corsi : Tabella</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Nome campo</th> <th>Tipo dati</th> </tr> </thead> <tbody> <tr> <td> Codice</td> <td>Testo</td> </tr> <tr> <td>Descrizione</td> <td>Testo</td> </tr> <tr> <td>DataInizio</td> <td>Data/ora</td> </tr> <tr> <td>DataFine</td> <td>Data/ora</td> </tr> </tbody> </table> </div>	Nome campo	Tipo dati	Codice	Testo	Descrizione	Testo	DataInizio	Data/ora	DataFine	Data/ora	<p style="text-align: center;">Nome query: CreaTabellaCorsi</p> <pre> CREATE TABLE Corsi (Codice CHAR(6), Descrizione CHAR(100) NOT NULL, DataInizio DATE NOT NULL, DataFine DATE NOT NULL, PRIMARY KEY(Codice)) </pre>		
Nome campo	Tipo dati												
Codice	Testo												
Descrizione	Testo												
DataInizio	Data/ora												
DataFine	Data/ora												
<div style="border: 1px solid black; padding: 5px;"> <p>Allievi : Tabella</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Nome campo</th> <th>Tipo dati</th> </tr> </thead> <tbody> <tr> <td> Codice</td> <td>Testo</td> </tr> <tr> <td>Cognome</td> <td>Testo</td> </tr> <tr> <td>Nome</td> <td>Testo</td> </tr> <tr> <td>Telefono</td> <td>Testo</td> </tr> <tr> <td>Email</td> <td>Testo</td> </tr> </tbody> </table> </div>	Nome campo	Tipo dati	Codice	Testo	Cognome	Testo	Nome	Testo	Telefono	Testo	Email	Testo	<p style="text-align: center;">Nome query: CreaTabellaAllievi</p> <pre> CREATE TABLE Allievi (Codice CHAR(6), Cognome CHAR(20) NOT NULL, Nome CHAR(20) NOT NULL, Telefono CHAR(15) NOT NULL, Email CHAR(30), PRIMARY KEY(Codice)) </pre>
Nome campo	Tipo dati												
Codice	Testo												
Cognome	Testo												
Nome	Testo												
Telefono	Testo												
Email	Testo												

Tabelle di base		Codice SQL nella modalità SQL Visualizzazione SQL															
<p>ComposizioneCorsi : Tabella</p> <table border="1"> <thead> <tr> <th>Nome campo</th> <th>Tipo dati</th> </tr> </thead> <tbody> <tr> <td>Allievo</td> <td>Testo</td> </tr> <tr> <td>Corso</td> <td>Testo</td> </tr> <tr> <td>Livello</td> <td>Numerico</td> </tr> <tr> <td>SuperatoTestFinale</td> <td>Sì/No</td> </tr> </tbody> </table>	Nome campo	Tipo dati	Allievo	Testo	Corso	Testo	Livello	Numerico	SuperatoTestFinale	Sì/No	<p>Nome query: CreaTabellaComposizioneCorsi</p> <pre>CREATE TABLE ComposizioneCorsi (Allievo CHAR(6), Corso CHAR(6), Livello INTEGER, SuperatoTestFinale BIT, PRIMARY KEY(Allievo, Corso), FOREIGN KEY(Corso) REFERENCES Corsi(Codice), FOREIGN KEY(Allievo) REFERENCES Allievi(Codice))</pre>						
Nome campo	Tipo dati																
Allievo	Testo																
Corso	Testo																
Livello	Numerico																
SuperatoTestFinale	Sì/No																
<p>Domande : Tabella</p> <table border="1"> <thead> <tr> <th>Nome campo</th> <th>Tipo dati</th> </tr> </thead> <tbody> <tr> <td>Codice</td> <td>Numerico</td> </tr> <tr> <td>Quesito</td> <td>Testo</td> </tr> <tr> <td>Risposta1</td> <td>Testo</td> </tr> <tr> <td>Risposta2</td> <td>Testo</td> </tr> <tr> <td>Risposta3</td> <td>Testo</td> </tr> <tr> <td>Risposta4</td> <td>Testo</td> </tr> <tr> <td>RispostaEsatta</td> <td>Numerico</td> </tr> </tbody> </table>	Nome campo	Tipo dati	Codice	Numerico	Quesito	Testo	Risposta1	Testo	Risposta2	Testo	Risposta3	Testo	Risposta4	Testo	RispostaEsatta	Numerico	<p>Nome query: CreaTabellaDomande</p> <pre>CREATE TABLE Domande (Codice INTEGER, Quesito CHAR(255) NOT NULL, Risposta1 CHAR(100) NOT NULL, Risposta2 CHAR(100) NOT NULL, Risposta3 CHAR(100) NOT NULL, Risposta4 CHAR(100) NOT NULL, RispostaEsatta INTEGER NOT NULL, PRIMARY KEY(Codice))</pre>
Nome campo	Tipo dati																
Codice	Numerico																
Quesito	Testo																
Risposta1	Testo																
Risposta2	Testo																
Risposta3	Testo																
Risposta4	Testo																
RispostaEsatta	Numerico																
<p>Test : Tabella</p> <table border="1"> <thead> <tr> <th>Nome campo</th> <th>Tipo dati</th> </tr> </thead> <tbody> <tr> <td>Codice</td> <td>Testo</td> </tr> <tr> <td>Descrizione</td> <td>Testo</td> </tr> </tbody> </table>	Nome campo	Tipo dati	Codice	Testo	Descrizione	Testo	<p>Nome query: CreaTabellaTest</p> <pre>CREATE TABLE Test (Codice CHAR(6), Descrizione CHAR(255) NOT NULL, PRIMARY KEY(Codice))</pre>										
Nome campo	Tipo dati																
Codice	Testo																
Descrizione	Testo																
<p>ComposizioneTest : Tabella</p> <table border="1"> <thead> <tr> <th>Nome campo</th> <th>Tipo dati</th> </tr> </thead> <tbody> <tr> <td>Test</td> <td>Testo</td> </tr> <tr> <td>Domanda</td> <td>Numerico</td> </tr> </tbody> </table>	Nome campo	Tipo dati	Test	Testo	Domanda	Numerico	<p>Nome query: CreaTabellaComposizioneTest</p> <pre>CREATE TABLE ComposizioneTest (Test CHAR(6), Domanda INTEGER, PRIMARY KEY(Test, Domanda), FOREIGN KEY(Test) REFERENCES Test(Codice), FOREIGN KEY(Domanda) REFERENCES Domande(Codice))</pre>										
Nome campo	Tipo dati																
Test	Testo																
Domanda	Numerico																
<p>Risultati : Tabella</p> <table border="1"> <thead> <tr> <th>Nome campo</th> <th>Tipo dati</th> </tr> </thead> <tbody> <tr> <td>Allievo</td> <td>Testo</td> </tr> <tr> <td>Test</td> <td>Testo</td> </tr> <tr> <td>Corso</td> <td>Testo</td> </tr> <tr> <td>Valutazione</td> <td>Numerico</td> </tr> </tbody> </table>	Nome campo	Tipo dati	Allievo	Testo	Test	Testo	Corso	Testo	Valutazione	Numerico	<p>Nome query: CreaTabellaRisultati</p> <pre>CREATE TABLE Risultati (Allievo CHAR(6), Test CHAR(6), Corso CHAR(6), Valutazione DOUBLE, PRIMARY KEY(Allievo, Test, Corso), FOREIGN KEY(Allievo) REFERENCES Allievi(Codice), FOREIGN KEY(Test) REFERENCES Test(Codice))</pre>						
Nome campo	Tipo dati																
Allievo	Testo																
Test	Testo																
Corso	Testo																
Valutazione	Numerico																

Lo schema logico relazionale del database in Access diventa quello che segue.



Le **interrogazioni sul database** sono realizzate mediante questi codici sorgente SQL che devono essere eseguiti in un DBMS.

Interrogazione: perfezionamento iscrizione

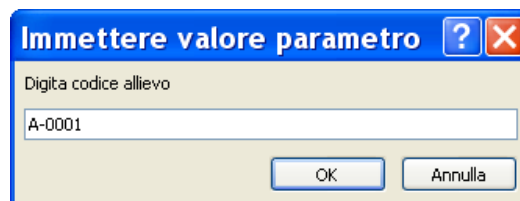
Perfezionare l'iscrizione di un allievo sulla base del risultato raggiunto al test (inserimento del livello corrispondente).

Dopo aver eseguito il test di ingresso on-line, una pagina di accesso ai dati memorizza il risultato (colonna *Valutazione*) raggiunto da un allievo nella tabella *Risultati*. Il perfezionamento dell'iscrizione può essere quindi effettuato in modo interattivo sulla base della procedura seguente.

1. Verifica del risultato raggiunto da un allievo, noto il suo codice, eseguendo la seguente query parametrica (di nome *RisultatoAllievo*).

```
SELECT Risultati.Test, Risultati.Corso, Risultati.Valutazione
FROM Allievi INNER JOIN Risultati
ON Allievi.Codice = Risultati.Allievo
WHERE Allievi.Codice= [Digita codice allievo]
```

Nel codice SQL, *[Digita codice allievo]* è un parametro di ingresso il cui valore viene richiesto da Access prima dell'esecuzione dell'interrogazione con una finestra di dialogo di questo tipo.



2. Aggiornamento della tabella *ComposizioneCorsi*, inserendo il livello corrispondente al risultato ottenuto (calcolato manualmente), con la seguente query *UPDATE* (di nome *ModificaLivello*).

```
UPDATE ComposizioneCorsi
SET Livello = [Digita valore livello]
WHERE Allievo=[Digita codice allievo] AND
Corso = [Digita codice corso]
```

L'esecuzione della query di aggiornamento richiede all'operatore il livello raggiunto (parametro *[Digita valore livello]*) da un allievo per un certo corso *[Digita codice corso]*) e il suo codice (parametro *[Digita codice allievo]*), per poi effettuare la modifica nella riga corrispondente nella tabella *ComposizioneCorsi*.

Ad esempio, possiamo ipotizzare la seguente corrispondenza tra il campo *Valutazione* raggiunto nel test (tabella *Risultati*) e la colonna *Livello* (tabella *ComposizioneCorsi*):

- *Valutazione* minore del 60%, *Livello*=1;
- *Valutazione* maggiore o uguale al 60% e minore dell'80%, *Livello*=2;
- *Valutazione* maggiore o uguale dell'80%, *Livello*=3.

In alternativa, in alcuni DBMS (non in Access), è possibile effettuare le due operazioni precedenti in modo automatico associando alla tabella *Risultati* un trigger eseguito in risposta a ogni operazione di inserimento. Un trigger di inserimento è definito da uno script SQL, il cui codice è eseguito dal DBMS per ogni comando *INSERT*.

Nel nostro progetto, lo script SQL del trigger si presenta con una forma del tipo seguente. Nel codice, *Inserted* rappresenta una tabella dinamica con la riga inserita nella tabella *Risultati* che provoca l'esecuzione del trigger. La sintassi potrebbe essere differente nei diversi DBMS.

```
CREATE TRIGGER ModificaLivello
ON Risultati
FOR INSERT
IF (SELECT Test FROM Inserted LIKE 'TING$') AND
   (SELECT Valutazione FROM Inserted < 60.0)
BEGIN
    UPDATE ComposizioneCorsi
    SET Livello = 1
    WHERE Corso = (SELECT Corso FROM Inserted) AND
           Allievo = (SELECT Allievo FROM Inserted)
END
ELSE
IF (SELECT Test FROM Inserted LIKE 'TING$') AND
   (SELECT Valutazione FROM Inserted >= 60.0) AND
   (SELECT Valutazione FROM Inserted < 80.0)
BEGIN
    UPDATE ComposizioneCorsi
    SET Livello = 2
    WHERE Corso = (SELECT Corso FROM Inserted) AND
           Allievo = (SELECT Allievo FROM Inserted)
END
ELSE
BEGIN
    UPDATE ComposizioneCorsi
    SET Livello = 3
    WHERE Corso = (SELECT Corso FROM Inserted) AND
           Allievo = (SELECT Allievo FROM Inserted)
END
END
```

Interrogazione: elenco allievi

Stampare l'elenco degli allievi che hanno superato il test finale.

Questo servizio può essere realizzato con la seguente interrogazione di nome *ElencoAllievi*.

```
SELECT Allievi.Cognome, Allievi.Nome, Allievi.Telefono
FROM Allievi INNER JOIN ComposizioneCorsi
ON Allievi.Codice = ComposizioneCorsi.Allievo
WHERE ComposizioneCorsi.Corso = [Digita codice corso] AND
      ComposizioneCorsi.SuperatoTestFinale = TRUE
```

Realizzare le pagine Web idonee a gestire la somministrazione del test d'ingresso e l'eventuale interazione con la base di dati

Questa operazione richiede la realizzazione di pagine di accesso ai dati del database *DBCorsiOnLine*. Nel seguito:

- ipotizzeremo che il server Web coincida con il server di database. Nel nostro progetto è sufficiente che nel server Web siano installati i driver per il RDBMS Access;
- useremo come tecnologia per la creazione di pagine di accesso ai dati l'ASP basato sul linguaggio di scripting VBScript.

Questa operazione può essere realizzata completando i seguenti passi operativi.

1. Creare una pagina Web, di nome *RichiamaTest.htm*, con un form in cui l'allievo deve digitare il proprio codice, il codice del corso e quello del test che deve eseguire. Questa pagina HTML richiama la pagina ASP, *EseguiTest.asp*, che esegue una query nel database per ricercare le domande del test e visualizzarle in una nuova pagina HTML dinamica.

I nomi delle caselle di testo sono quelli riportati nella figura precedente, mentre il form richiama la pagina *EseguiTest.asp* mediante la seguente struttura del tag doppio `<FORM>`, inserita nel corpo della pagina Web.

```
<FORM METHOD="POST" ACTION="EseguiTest.asp">
...
</FORM>
```

Pagina ASP *EseguiTest.asp*

```
<%@ Language=VBScript %>
<% Response.Buffer=True %>

<HTML>
<HEAD><TITLE>Esegui test d'ingresso</TITLE></HEAD>
<BODY>
<H3>Test d'ingresso</H3>
<% Dim ConnessioneDB, rsTabella, QuerySQL, Test, Allievo, Corso, i
'Apertura della connessione al database
Set ConnessioneDB = Server.CreateObject("ADODB.Connection")
ConnessioneDB.Open "DRIVER={Microsoft Access Driver (*.mdb)}; " &_
"DBQ=" & Server.MapPath("DBCorsiOnLine.mdb")
```



```

'Raccolta dei dati digitati dall'utente nel browser
Test = Request.Form("txtTest")
Allievo = Request.Form("txtAllievo")
Corso = Request.Form("txtCorso")
'Salvataggio dati in un cookie
Response.Cookies("Dati")("Test") = Test
Response.Cookies("Dati")("Allievo") = Allievo
Response.Cookies("Dati")("Corso") = Corso

'Definizione codice SQL della query per ricercare le domande di un test
QuerySQL="SELECT Domande.* " &_
        "FROM Domande INNER JOIN " &_
        "(Test INNER JOIN ComposizioneTest ON " &_
        "Test.Codice = ComposizioneTest.Test) " &_
        "ON Domande.Codice = ComposizioneTest.Domanda " &_
        "WHERE Test.Codice ='" & Test & "'";"
Set rsTabella = ConnessioneDB.Execute(QuerySQL)

'Creazione di una pagina HTML dinamica con i test.
'Le domande sono individuate da un nome del tipo NAME=i, dove i e' il numero
'della domanda. Il test e' indipendente dal numero di domande memorizzato.
'Dopo la compilazione del test, la pagina richiama CalcolaRisultato.asp
' che effettua il calcolo del risultato raggiunto.
%>
<% If rsTabella.EOF=True And rsTabella.EOF=True Then %>
  <H4>Non ci dati disponibili!</H4>
<% Else %>
  <FORM METHOD="POST" ACTION="CalcolaRisultato.asp">
  <%i=1
  Do until rsTabella.EOF%>
    <P><%=rsTabella.Fields("Quesito")%><BR>
    <%=rsTabella.Fields("Risposta1")%>
    <INPUT TYPE="RADIO" NAME="<%=i%>" VALUE="1"><BR>
    <%=rsTabella.Fields("Risposta2")%>
    <INPUT TYPE="RADIO" NAME="<%=i%>" VALUE="2"><BR>
    <%=rsTabella.Fields("Risposta3")%>
    <INPUT TYPE="RADIO" NAME="<%=i%>" VALUE="3"><BR>
    <%=rsTabella.Fields("Risposta4")%>
    <INPUT TYPE="RADIO" NAME="<%=i%>" VALUE="4"><BR>
  </P>
  <% i = i+ 1
  rsTabella.MoveNext %>
<% Loop %>
  <P>
    <INPUT TYPE="SUBMIT" NAME="submit" VALUE="Invia le risposte del test">
  </P>
</FORM>
<% End If %>

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

```

La pagina ASP produce un risultato del tipo descritto nella figura della pagina seguente.

Indirizzo <http://localhost/CorsiOnLine/default.htm> Vai

Home

Test d'ingresso

test ingresso quesito 1

risposta 1.1

risposta 1.2

risposta 1.3

risposta 1.4

test ingresso quesito 2

risposta 2.1

risposta 2.2

risposta 2.3

risposta 2.4

test ingresso quesito 3

risposta 3.1

risposta 3.2

risposta 3.3

risposta 3.4

Invia le risposte del test

2. La pagina *EseguiTest.asp* richiama un'altra pagina ASP, *CalcolaRisultato.asp*, che calcola il risultato raggiunto dall'allievo nel test e, dopo avere visualizzato il risultato in una pagina HTML dinamica, aggiorna la tabella *Risultati* con la percentuale ottenuta nel test d'ingresso.

Pagina ASP *CalcolaRisultato.asp*

```
<%@ Language=VBScript %>
<% Response.Buffer=True %>

<%
Dim ConnessioneDB, rsTabella, QuerySQL
Dim Test, Allievo, Corso
Dim i, RisposteOK, RispostaData, RecordModificati

'Apertura della connessione al database
Set ConnessioneDB = Server.CreateObject("ADODB.Connection")
ConnessioneDB.Open "DRIVER={Microsoft Access Driver (*.mdb)}; " & _
    "DBQ=" & Server.MapPath("DBCorsiOnLine.mdb")

'Letture dei dati memorizzati nei cookie.
Test = Request.Cookies("Dati")("Test")
Allievo = Request.Cookies("Dati")("Allievo")
Corso = Request.Cookies("Dati")("Corso")

'Definizione query SQL per la lettura della risposta esatta nel test scelto.
QuerySQL="SELECT Domande.RispostaEsatta " & _
    "FROM Domande INNER JOIN " & _
    "(Test INNER JOIN ComposizioneTest ON " & _
    "Test.Codice = ComposizioneTest.Test) " & _
    "ON Domande.Codice = ComposizioneTest.Domanda " & _
    "WHERE Test.Codice ='" & Test & "'"
Set rsTabella = ConnessioneDB.Execute(QuerySQL)
```

```
'Calcolo del numero di risposte esatte.
i=1
RisposteOK = 0
Do Until rsTabella.EOF
    RispostaEsatta=rsTabella.Fields("RispostaEsatta")
    SELECT CASE Request.Form("&i &")
    CASE "1"
        RispostaData=1
    CASE "2"
        RispostaData=2
    CASE "3"
        RispostaData=3
    CASE "4"
        RispostaData=4
    END SELECT
    IF RispostaEsatta=RispostaData THEN
        RisposteOK = RisposteOK +1
    END IF
    i = i +1
    rsTabella.MoveNext
Loop

'Calcolo della percentuale di domande corrette.
Risultato = (RisposteOK/(i-1))*100

'Esecuzione della query di aggiornamento per aggiornare la tabella
' con il risultato del test.
QueryString="UPDATE Risultati " &_
    "SET Valutazione=" & Round(Risultato,0) &_
    " WHERE Allievo='" & Allievo & "' AND " &_
    " Test='" & Test & "' AND " &_
    " Corso='" & Corso & "'"

ConnessioneDB.Execute QueryString, RecordModificati

'Stampa del risultato all'utente
Response.Write("<H3>Risultato raggiunto nel test d'ingresso:<BR>")
Response.Write(Round(Risultato,0) & " %</H3>")

rsTabella.Close : Set rsTabella = Nothing
ConnessioneDB.Close : Set ConnessioneDB = Nothing
%>
```

NOTE per l'uso del sito Web. In allegato a questa soluzione, è possibile eseguire il download della cartella con il sito Web che contiene l'applicazione ASP appena realizzata. Il sito è pronto per essere pubblicato (in locale o in Internet) in un computer in cui è stato installato il servizio WWW. Il testing è stato effettuato pubblicando il sito in un computer locale con Windows XP Professional in cui è stato installato il servizio WWW, creato con il programma di sistema IIS (*Internet Information Services*). Dopo la pubblicazione (sia in locale sia in Internet), affinché funzionino le operazioni di aggiornamento sulla base di dati, è necessario che il file di database *DBCorsiOnLine.mdb*, nella home folder del sito, disponga dei necessari permessi di scrittura (nel sito www.tramontana.it consulta anche *Impostazione dei permessi di scrittura a un file di database con Windows XP* nella sezione *Approfondimenti ICT* di ICT Web).