

Esame di Stato Istituto Tecnico Industriale CORSO DI ORDINAMENTO Indirizzo: INFORMATICA

Tema di:

INFORMATICA GENERALE E APPLICAZIONI TECNICO-SCIENTIFICHE

Anno Scolastico: 2007-2008

Note per lo studente

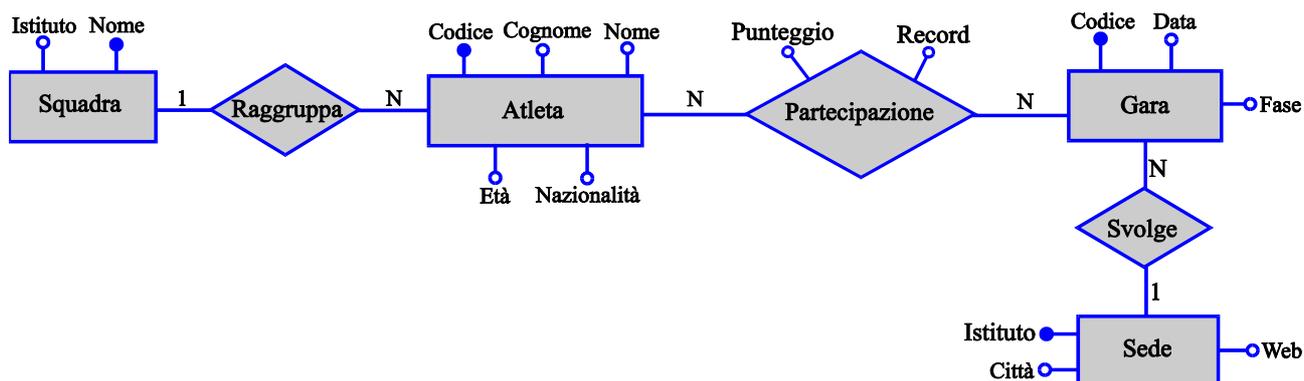
In allegato a questa prova è possibile effettuare il download del file di database **DBOlimpiadiInfo.mdb**, realizzato in Access 2003 che contiene:

1. le query di definizione dei dati per la creazione delle tabelle del database;
2. le tabelle di base e le relazioni del database;
3. le interrogazioni richieste;
4. un insieme di dati inseriti nelle tabelle di base per il testing dei servizi del sistema informativo.

La progettazione di un database non ha un'unica soluzione. Quella proposta può essere quindi variata in base all'esperienza del programmatore.

■■■ PROGETTO CONCETTUALE

SCHEMA ER



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

- R1. L'attributo *Codice* di un *Atleta* può assumere il formato AXXXXX.
- R2. L'attributo *Record* di una *Partecipazione* rappresenta, se uguale a *TRUE*, un punteggio record in una gara.
- R3. L'attributo *ID* di una *Gara* è un contatore (numero intero che si incrementa automaticamente).
- R4. L'attributo *Fase* di una *Gara* può assumere solo i valori: "scolastica", "regionale", "nazionale", "internazionale".
- R5. L'attributo *Web* di una *Sede* memorizza l'URL del sito Web dell'istituto che ospita la gara.

■■■ 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 *Squadra* della tabella *Atleti* è in relazione con la tabella *Squadre* mediante la chiave primaria *Nome*.
- V2. La chiave esterna *Atleta* della tabella *Partecipazioni* è in relazione con la tabella *Atleti* mediante la chiave primaria *Codice*.
- V3. La chiave esterna *Gara* della tabella *Partecipazioni* è in relazione con la tabella *Gare* mediante la chiave primaria *ID*.
- V4. La chiave esterna *Sede* della tabella *Gare* è in relazione con la tabella *Sedi* mediante la chiave primaria *Istituto*.

CODIFICA

Per tradurre nel DBMS lo **schema logico del database** si deve eseguire, una sola volta, questo codice sorgente SQL.

```
CREATE DATABASE DBolimpiadiInfo

CREATE TABLE Squadre
(
  Nome      CHAR(20),
  Istituto CHAR(30) NOT NULL,
  PRIMARY KEY(Nome)
)

CREATE TABLE Atleti
(
  Codice     CHAR(6) CHECK(Codice LIKE 'A_____'),
  Cognome    CHAR(20) NOT NULL,
  Nome       CHAR(20) NOT NULL,
  Et         INTEGER NOT NULL,
  Squadra    CHAR(20),
  Nazionalit  CHAR(20) NOT NULL,
  PRIMARY KEY(Codice),
  FOREIGN KEY(Squadra) REFERENCES Squadre(Nome)
)

CREATE TABLE Sedi
(
  Istituto CHAR(30),
  Citt      CHAR(20),
  Web       CHAR(30) NOT NULL,
  PRIMARY KEY(Istituto)
)

CREATE TABLE Gare
(
  ID      INTEGER IDENTITY(1,1),
  Data    DATE NOT NULL,
  Sede    CHAR(30),
  Fase    CHAR(20) CHECK(Fase IN ('scolastica', 'regionale', 'nazionale',
                                'internazionale') NOT NULL),
  PRIMARY KEY(ID),
  FOREIGN KEY(Sede) REFERENCES Sedi(Istituto)
)

CREATE TABLE Partecipazioni
(
  Atleta     CHAR(6),
  Gara       INTEGER,
  Punteggio  INTEGER NOT NULL,
```

```
Record    BIT DEFAULT 0,
PRIMARY KEY(Atleta, Gara),
FOREIGN KEY(Atleta) REFERENCES Atleti(Codice),
FOREIGN KEY(Gara)   REFERENCES Gare(ID)
)
```

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

1^a interrogazione

Stampare l'elenco degli atleti raggruppati per squadre per ogni singola fase

```
SELECT Atleti.Cognome, Atleti.Nome, Squadre.Nome, Squadre.Istituto
FROM Squadre, Atleti, Partecipazioni, Gare
WHERE Squadre.Nome = Atleti.Squadra AND
      Atleti.Codice = Partecipazioni.Atleta AND
      Partecipazioni.Gara = Gare.ID AND
      Gare.Fase = [Digita fase delle gare]
ORDER BY Squadre.Istituto, Squadre.Nome
```

2^a interrogazione

Dato il nome di un atleta stampare i risultati ottenuti nelle diverse gare alle quali ha partecipato

```
SELECT Partecipazioni.Punteggio, Gare.Sede, Gare.Fase
FROM Atleti, Partecipazioni, Gare
WHERE Atleti.Codice = Partecipazioni.Atleta AND
      Partecipazioni.Gara = Gare.ID AND
      Atleti.Cognome = [Digita cognome atleta] AND
      Atleti.Nome = [Digita nome atleta]
```

3^a interrogazione

Stampare il calendario delle gare

```
SELECT *
FROM Gare
```

4^a interrogazione

Stampare una scheda informativa (cognome, nome, istituto scolastico di provenienza, nazionalità) del vincitore e della squadra vincitrice

```
SELECT Atleti.Cognome, Atleti.Nome, Squadre.Istituto, Atleti.Nazionalità
FROM Squadre, Atleti, Partecipazioni, Gare
WHERE Squadre.Nome = Atleti.Squadra AND
      Atleti.Codice = Partecipazioni.Atleta AND
      Partecipazioni.Gara = Gare.ID AND
      Gare.ID = [Digita ID della gara] AND
      Partecipazioni.Punteggio = (SELECT MAX(Partecipazioni.Punteggio)
                                  FROM Partecipazioni INNER JOIN Gare
                                  ON Partecipazioni.Gara = Gare.ID
                                  WHERE Gare.ID = [Digita ID della gara] )
```

5^a interrogazione

Stampare la classifica per ciascuna gara (a parità di punteggio vengono privilegiati gli atleti più giovani)

```
SELECT Atleti.Cognome, Atleti.Nome, Partecipazioni.Punteggio, Atleti.Età,
      Squadre.Istituto
FROM Squadre, Atleti, Partecipazioni, Gare
WHERE Squadre.Nome = Atleti.Squadra AND
      Atleti.Codice = Partecipazioni.Atleta AND
      Partecipazioni.Gara = Gare.ID AND
      Gare.ID = [Digita ID della gara]
ORDER BY Partecipazioni.Punteggio DESC, Atleti.Età ASC
```

6ª interrogazione

Aggiornare, per ciascuna fase (scolastica-regionale-nazionale-internazionale) gli eventuali punteggi regord

```
UPDATE Partecipazioni
SET Partecipazioni.Record = TRUE
WHERE Partecipazioni.Punteggio = (SELECT MAX(Partecipazioni.Punteggio)
FROM Partecipazioni INNER JOIN Gare
ON Partecipazioni.Gara = Gare.ID
WHERE Gare.Fase = [Digita fase gare] )
```

7ª interrogazione

Calcolare il punteggio medio ottenuto durante la prima selezione, per ciascun istituto scolastico

```
SELECT Gare.Sede, AVG(Partecipazioni.Punteggio) AS 'Punteggio medio'
FROM Partecipazioni INNER JOIN Gare
ON Partecipazioni.Gara = Gare.ID
WHERE Gare.Fase = 'scolastica'
GROUP BY Gare.Sede
```

8ª interrogazione

Stampare per ciascuna squadra il numero di “atleti” partecipanti e l’età media

```
SELECT Squadre.Nome, COUNT(*) AS 'Numero atleti', AVG(Atleti.Età) AS 'Età media'
FROM Squadre INNER JOIN Atleti
ON Squadre.Nome = Atleti.Squadra
GROUP BY Squadre.Nome
```

Note per lo studente

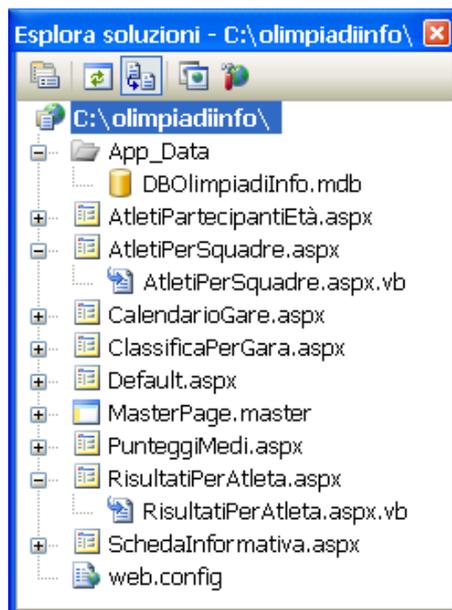
Il sito Web è stato realizzato in ASP.NET con il framework 3.5, per fornire la possibilità a tutti gli allievi di collaudare il sistema proposto. In allegato a questa prova è possibile effettuare il download dell’intero sito Web, memorizzato nella cartella *olimpiadinfo*. Il sito Web è completo e funzionante per cui può essere collaudato pubblicandolo su un server Web di Internet oppure dell’intranet del laboratorio del proprio Istituto. Le pagine *.aspx* del sito accedono al file di database di Access *DBInformaticaInfo.mdb*, con la soluzione del problema.

CREAZIONE DEL SITO WEB

Il sito è basato sulla pagina master *MasterPage.master* che fornisce il layout di tutte le pagine del sito. La **struttura logica** del sito è quella descritta nella figura seguente.



La **struttura fisica** del sito Web è illustrata nella figura successiva.



SVILUPPO DELL'APPLICAZIONE CLIENT-SERVER DISTRIBUITA

Nel seguito realizzeremo le prime due operazioni del sito Web. Gli algoritmi realizzati nelle altre operazioni sono analoghi a quelli che descriveremo.

Operazione: stampare l'elenco degli atleti raggruppati per squadre per ogni singola fase

Per realizzare questa operazione, progettiamo la GUI del Web form *AtletiPerSquadra.aspx*, che include il file *AtletiPerSquadra.aspx.vb* con il codice sorgente che esegue la prima query sul database.

- File *AtletiPerSquadra.aspx* nella modalità progettazione con la GUI

Scegli la fase	<input type="text" value="scolastica"/> <input type="text" value="regionale"/> <input type="text" value="nazionale"/> <input type="text" value="internazionale"/>	IstFasi
Column0	Column1	Column2
abc	abc	abc

- File *AtletiPerSquadra.aspx.vb*

```
Imports System.Data.OleDb
Imports System.Data
```

```
Partial Class AtletiPerSquadre
    Inherits System.Web.UI.Page
```

```
    Private percorsoDB As String=Me.Server.MapPath("App_Data/DBOlimpiadiInfo.mdb")
    Private stringaConnDB As String = "Provider=Microsoft.Jet.OLEDB.4.0; " & _
        "Data Source=" & percorsoDB
```

```

Private connDB As OleDbConnection

Protected Sub Page_Load(ByVal sender As Object, ByVal e As EventArgs) _
    Handles Me.Load
    lstFasi.AutoPostBack = True
End Sub

Protected Sub lstFasi_SelectedIndexChanged(ByVal sender As Object, _
    ByVal e As EventArgs) Handles lstFasi.SelectedIndexChanged
    AggiornaTabella()
End Sub

Sub AggiornaTabella()
    Try
        connDB = New OleDbConnection(stringaConnDB) ' connessione database
        Dim oggettoDB As New DataSet, codiceSQL As String
        codiceSQL = _
            "SELECT Atleti.Cognome, Atleti.Nome, Squadre.Nome, " & _
            "    Squadre.Istituto " & _
            " FROM Squadre, Atleti, Partecipazioni, Gare " & _
            " WHERE Squadre.Nome = Atleti.Squadra AND " & _
            "    Atleti.Codice = Partecipazioni.Atleta AND " & _
            "    Partecipazioni.Gara = Gare.ID AND " & _
            "    Gare.Fase = '" & _
            lstFasi.Items(lstFasi.SelectedIndex).ToString() & "' " & _
            "    ORDER BY Squadre.Istituto, Squadre.Nome"
        Dim adattatoreDati As New OleDbDataAdapter(codiceSQL, connDB)
        adattatoreDati.Fill(oggettoDB, "Tabella")
        grdGriglia.DataSource = oggettoDB.Tables("Tabella")
        Me.DataBind()
    Catch eccezione As Exception
        connDB.Close() ' chiusura connessione database in caso di errore
    Finally
        If (connDB.State = ConnectionState.Open) Then
            connDB.Close()
        End If
    End Try
End Sub
End Class

```

Operazione: dato il nome di un atleta stampare i risultati ottenuti nelle diverse gare alle quali ha partecipato

Per realizzare questa operazione, progettiamo la GUI del Web form *RisultatiPerAtleta.aspx*, che include il file *RisultatiPerAtleta.aspx.vb* con il codice sorgente che esegue la seconda query sul database.

- File *RisultatiPerAtleta.aspx* nella modalità progettazione con la GUI

Cognome	<input type="text"/>	txtCognome
Nome	<input type="text"/>	txtNome
<input type="button" value="Ricerca"/>		
Column0	Column1	Column2
abc	abc	abc
abc	abc	grdGriglia
abc	abc	abc
abc	abc	abc
abc	abc	abc

■ File *RisultatiPerAtleta.aspx.vb*

```
Imports System.Data.OleDb
Imports System.Data

Partial Class RisultatiPerAtleta
    Inherits System.Web.UI.Page
    Private percorsoDB As String=Me.Server.MapPath("App_Data/DBOlimpiadiInfo.mdb")
    Private stringaConnDB As String = "Provider=Microsoft.Jet.OLEDB.4.0; " & _
        "Data Source=" & percorsoDB

    Private connDB As OleDbConnection

    Protected Sub btnRicerca_Click(ByVal sender As Object, ByVal e As EventArgs) _
        Handles btnRicerca.Click
        AggiornaTabella()
    End Sub

    Sub AggiornaTabella()
        Try
            connDB = New OleDbConnection(stringaConnDB) ' connessione database
            Dim oggettoDB As New DataSet, codiceSQL As String
            codiceSQL = _
                "SELECT Partecipazioni.Punteggio, Gare.Sede, Gare.Fase " & _
                " FROM Atleti, Partecipazioni, Gare" & _
                " WHERE Atleti.Codice = Partecipazioni.Atleta AND " & _
                " Partecipazioni.Gara = Gare.ID AND " & _
                " Atleti.Cognome =' " & txtCognome.Text.ToString() & "' AND " & _
                " Atleti.Nome =' " & txtNome.Text.ToString() & "'"
            Dim adattatoreDati As New OleDbDataAdapter(codiceSQL, connDB)
            adattatoreDati.Fill(oggettoDB, "Tabella")
            grdGriglia.DataSource = oggettoDB.Tables("Tabella")
            Me.DataBind()
        Catch eccezione As Exception
            connDB.Close() ' chiusura connessione database in caso di errore
        Finally
            If (connDB.State = ConnectionState.Open) Then
                connDB.Close()
            End If
        End Try
    End Sub
End Class
```