

Esame di Stato Istituto Tecnico Commerciale
Proposta di soluzione della Seconda Prova
Indirizzo: PROGRAMMATORI
Tema di: INFORMATICA GENERALE E
APPLICAZIONI GESTIONALI

(Testo valevole anche per i corsi di ordinamento e per i corsi sperimentali dei progetti
 “SIRIO” programmatori e “MERCURIO”)

Anno Scolastico: 2006-2007

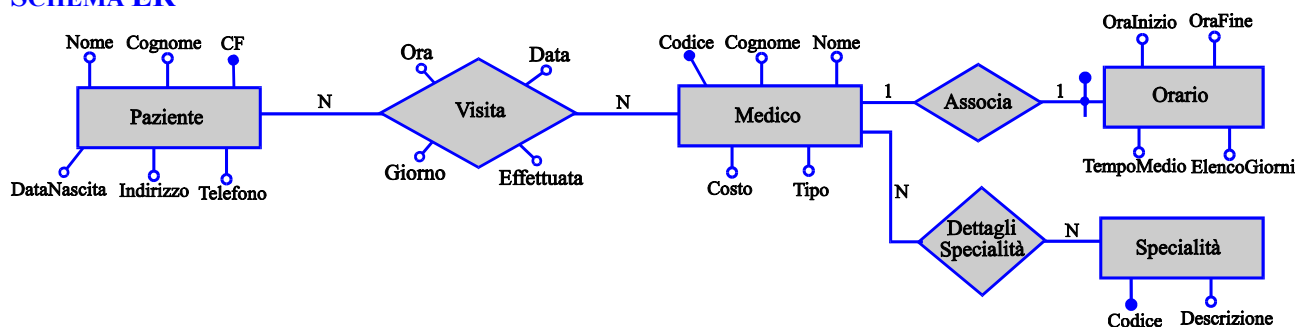
La progettazione di un database non ha un'unica soluzione. Quella proposta può essere quindi variata in base all'esperienza del programmatore.

La soluzione della prima **parte teorica** è descritta nel testo Tramontana “Sistemi informativi e SQL” della collana ICT, in particolare, per quanto riguarda:

- le procedure di backup, che sono illustrate a pagina 143 nella scheda integrativa “Il backup dei dati di un database”;
- la gestione della sicurezza, che è descritta a pagina 289 nell'appendice “Sicurezza e database”.

■■■ PROGETTO CONCETTUALE

SCHEMA ER

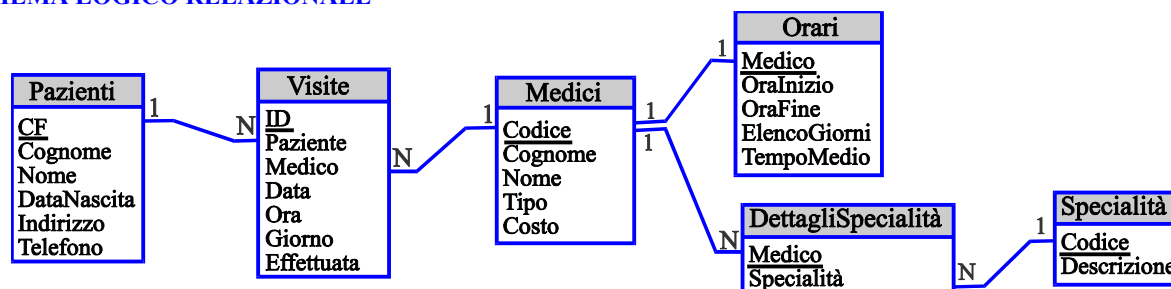


Lo schema ER deve seguire queste **regole**.

- R1. L'attributo *CF* di un *Paziente* memorizza il codice fiscale con 16 caratteri alfanumerici.
- R2. L'attributo *Giorno* di una *Visita* può assumere solo i valori: "lunedì", "martedì", "mercoledì", "giovedì", "venerdì" e "sabato".
- R3. L'attributo *Codice* di un *Medico* può assumere il formato M-XXXX.
- R4. L'attributo *Tipo* di un *Medico* può assumere solo i valori: "base" e "specialista".
- R5. L'attributo *ElencoGiorni* di un *Orario* memorizza un testo con la lista dei giorni in cui il medico è disponibile.
- R6. L'attributo *Codice* di una *Specialità* può assumere il formato S-XX.

■■■ 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 *Paziente* della tabella *Visite* è in relazione con la tabella *Pazienti* mediante la chiave primaria *CF*.
- V2. La chiave esterna *Medico* della tabella *Visite* è in relazione con la tabella *Medici* mediante la chiave primaria *Codice*.
- V3. La chiave esterna *Medico* della tabella *DettagliSpecialità* è in relazione con la tabella *Medici* mediante la chiave primaria *Codice*.
- V4. La chiave esterna *Specialità* della tabella *DettagliSpecialità* è in relazione con la tabella *Specialità* mediante la chiave primaria *Codice*.
- V5. La chiave esterna *Medico* della tabella *Orari* è in relazione con la tabella *Medici* mediante la chiave primaria *Codice*.

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

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

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

CODIFICA

Interrogazione 1: elenco giornaliero visite per medico

```
SELECT Visite.*
FROM Medici INNER JOIN Visite ON Medici.Codice = Visite.Medico
WHERE Medici.Codice = [Digita codice medico] AND
Visite.Data = [Digita data visite]
```

Interrogazione 2: elenco giornaliero visite prenotate e non effettuate

```
SELECT Visite.ID, Pazienti.Cognome, Medici.Cognome, Visite.Ora
FROM Pazienti, Visite, Medici
WHERE Pazienti.CF = Visite.Paziente AND
Visite.Medico = Medici.Codice AND
Visite.Effettuata = FALSE AND
Visite.Data = [Digita data visite]
```

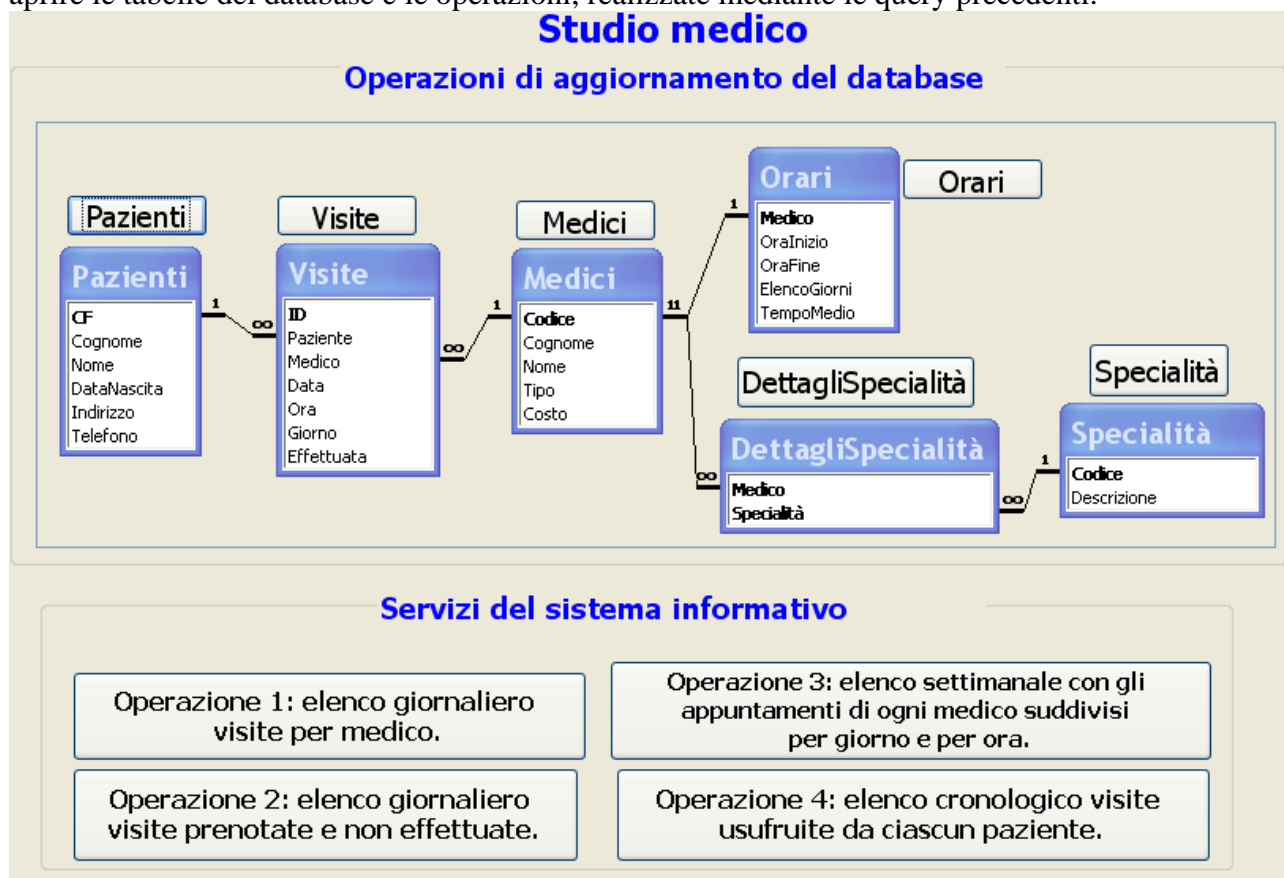
Interrogazione 3: elenco settimanale con gli appuntamenti di ogni medico suddivisi per giorno e per ora

```
SELECT Visite.Giorno, Visite.Ora
FROM Medici INNER JOIN Visite ON Medici.Codice = Visite.Medico
WHERE Visite.Effettuata = FALSE AND
Medici.Codice = [Digita codice medico]
GROUP BY Visite.Giorno, Visite.Ora
```

Interrogazione 4: elenco cronologico visite usufruite da ciascun paziente

```
SELECT Visite.*
FROM Pazienti INNER JOIN Visite ON Pazienti.CF = Visite.Paziente
WHERE Visite.Effettuata = TRUE AND
Pazienti.CF = [Digita codice fiscale paziente]
```

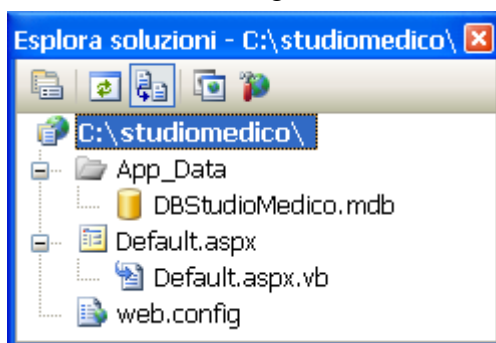
Il menu di scelta può anche essere realizzato con una maschera del RDBMS Access che permette di aprire le tabelle del database e le operazioni, realizzate mediante le query precedenti.



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 *studiomedico*. 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 *DBStudioMedico.mdb*, con la soluzione del problema.

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



SVILUPPO DELL'APPLICAZIONE CLIENT-SERVER DISTRIBUITA

Operazione: realizzare la pagina Web con la quale lo studio medico pubblicizza la propria attività fornendo l'indicazione dei servizi e il quadro orario

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

■ File *Default.aspx* nella modalità progettazione con la GUI

Servizi offerti (fai clic per visualizzare i nostri medici per ogni specialità):

Non associato

[IstSpecialità](#)

Medici		
Column0	Column1	Column2
abc	abc	abc
abc	grdMedici	abc
abc	abc	abc
abc	abc	abc
abc	abc	abc

txtCodice
 Digita il codice del medico per btnRicerca

	Column0	Column1	Column2
grdOrari	abc	abc	abc
	abc	abc	abc
	abc	abc	abc
	abc	abc	abc
	abc	abc	abc

■ File *Default.aspx.vb*

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

Partial Class _Default
    Inherits System.Web.UI.Page

    Private percorsoDB As String = Me.Server.MapPath("App_Data/DBStudioMedico.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
        IstSpecialità.AutoPostBack = True
        If Me.IsPostBack = False Then
            AggiornaListaSpecialità()
        End If
    End Sub

    Sub AggiornaListaSpecialità()
        Try
            connDB = New OleDbConnection(stringaConnDB) ' connessione database
            Dim oggettoDB As New DataSet, codiceSQL As String
            codiceSQL = "SELECT Descrizione FROM Specialità ORDER BY Descrizione"
            Dim adattatoreDati As New OleDbDataAdapter(codiceSQL, connDB)
            adattatoreDati.Fill(oggettoDB, "Tabella")
            IstSpecialità.DataSource = oggettoDB.Tables("Tabella")
            IstSpecialità.DataValueField = "Descrizione"
            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
```

```
Protected Sub lstSpecialità_SelectedIndexChanged(ByVal sender As Object, _
        ByVal e As EventArgs) Handles lstSpecialità.SelectedIndexChanged
    AggiornaMedici()
End Sub

Sub AggiornaMedici()
    Try
        connDB = New OleDbConnection(stringaConnDB) ' connessione database
        Dim oggettoDB As New DataSet, codiceSQL As String
        codiceSQL = _
            "SELECT Medici.* " & _
            " FROM Medici, DettagliSpecialità, Specialità " & _
            " WHERE Medici.Codice = DettagliSpecialità.Medico AND " & _
            "         DettagliSpecialità.Specialità = Specialità.Codice AND " & _
            "         Specialità.Descrizione ='" & _
            lstSpecialità.Items(lstSpecialità.SelectedIndex).ToString() & "'"
        Dim adattatoreDati As New OleDbDataAdapter(codiceSQL, connDB)
        adattatoreDati.Fill(oggettoDB, "Tabella")
        grdMedici.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

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

Sub AggiornaOrari()
    Try
        connDB = New OleDbConnection(stringaConnDB) ' connessione database
        Dim oggettoDB As New DataSet, codiceSQL As String
        codiceSQL = _
            "SELECT Orari.* " & _
            " FROM Orari " & _
            " WHERE Orari.Medico = '" & txtCodice.Text.ToString() & "'"
        Dim adattatoreDati As New OleDbDataAdapter(codiceSQL, connDB)
        adattatoreDati.Fill(oggettoDB, "Tabella")
        grdOrari.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
```