

## Simulazione seconda prova Esame di Stato

### Gestione Asl: alternanza scuola lavoro

(Parte prima)

*In un istituto superiore si vuole realizzare una base di dati per gestire il progetto di alternanza scuola lavoro. Possono partecipare al progetto tutti gli studenti delle classi III e IV.*

*Gli studenti all'inizio dell'anno scolastico possono scegliere, entro il 31 ottobre, tra periodi di tirocinio in alternanza scuola lavoro (Asl) presso aziende oppure dei moduli di approfondimento da frequentare in classe scelti tra le materie dell'anno scolastico che frequentano e offerti dall'Istituto in base al piano di offerta formativa (POF). Il periodo di tempo interessato per l'Asl o la frequenza in classe riguarda 15 giorni alla fine dell'anno scolastico e può quindi variare a seconda del calendario scelto dall'istituto.*

*Tutti gli studenti sono registrati con cognome, nome, data di nascita, corso frequentato, classe frequentata con indicazione della sezione. Ogni modulo di approfondimento viene identificato con codice, materia, titolo del corso, docente.*

*Le aziende interessate all'Asl vengono individuate con: denominazione, indirizzo, contatto, numero di telefono, fax, e-mail. Ogni azienda può richiedere più studenti.*

*Il sistema informativo deve essere realizzato in un calcolatore centrale (il server di database) al quale possono accedere i diversi terminali o PC (i computer client) installati presso gli uffici dell'intranet dell'istituto. Tutti i servizi del sistema informativo devono essere resi disponibili nel sito Web dell'intranet dell'istituto, in modo che il personale della scuola possa eseguire, in tempo reale, l'aggiornamento dei periodi di tirocinio e le operazioni sui dati disponibili. Il candidato definisca:*

- *lo schema concettuale e logico della base di dati;*
- *le relazioni della base di dati in linguaggio SQL.*

*Realizzi inoltre le seguenti interrogazioni.*

- *Dato uno studente di un certo anno visualizzare le sue scelte: Asl e/o moduli di approfondimento.*
- *Dato un modulo, l'elenco ordinato degli studenti che hanno chiesto di frequentarlo.*
- *Il numero di aziende che hanno richiesto studenti per l'Asl e il loro elenco.*
- *Il numero di studenti, raggruppati per classe, che hanno chiesto di effettuare il periodo di Asl.*

<b>Competenze verificate</b>	
■	Produrre il progetto concettuale di un sistema informativo
■	Tradurre un progetto concettuale nel database e nei servizi di un sistema informativo
■	Realizzare lo schema logico di un database relazionale mediante un codice sorgente SQL
■	Realizzare i servizi di un sistema informativo mediante query SQL

<b>Volumi di riferimento</b>		
<b>Teoria</b>	<b>Laboratorio</b>	
<b>ICT 3</b>	<b>ICT 3</b>	<b>Realizzazione di sistemi informativi. Access</b>
Modulo 8 - Sistemi informativi e database	Modulo 9 - SQL	Modulo 1 - Database e realizzazione dei servizi

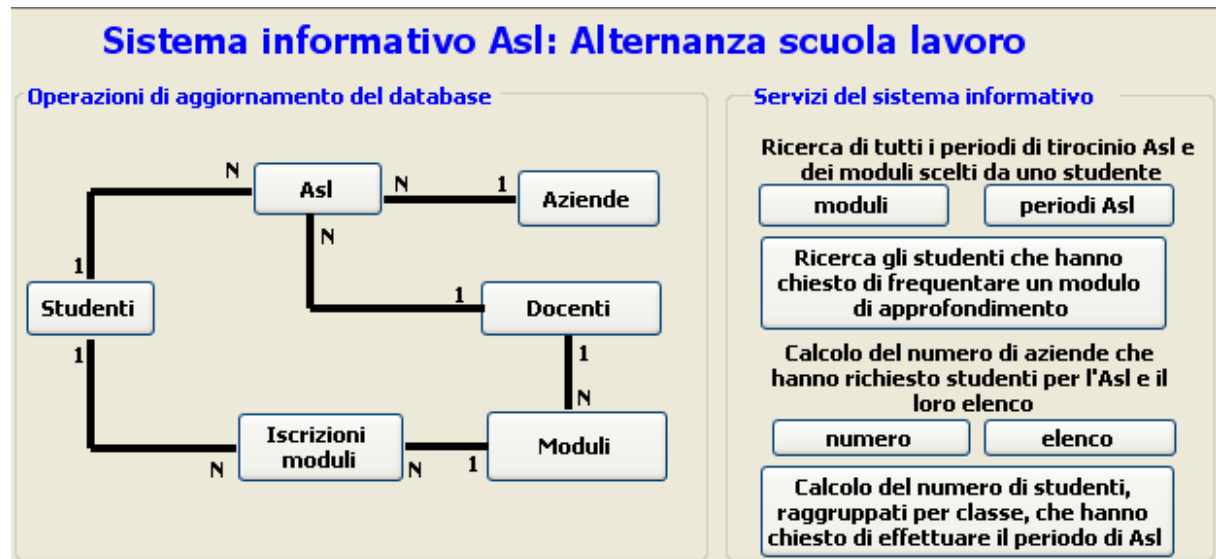
### Riflessioni per il docente

Il metodo di lavoro proposto ripercorre il ciclo di vita di un sistema informativo automatizzato attraverso l'analisi, il progetto, la realizzazione e il testing.

Il sistema informativo centralizzato è stato realizzato in Access impiegando l'implementazione dell'SQL nel RDBMS, per fornire la possibilità a tutti gli allievi di collaudare il sistema proposto.

In allegato a questa prova è anche possibile effettuare il download del file di database *Asl.mdb*, realizzato in Access 2003 (ma collaudato anche con Access 2000) che contiene:

1. le tabelle di base e le relazioni del database;
2. i servizi richiesti;
3. una maschera di avvio, che definisce (richiamando macro) una interfaccia grafica (riportata nella figura che segue) per l'utente al fine dell'aggiornamento dei dati nelle tabelle di base e dell'esecuzione delle operazioni del sistema;
4. un insieme di dati inseriti nelle tabelle di base per il testing dei servizi del sistema informativo.



## Analisi del sistema informativo

### Fase 1 - Specifiche per i dati

Il sistema deve raccogliere e memorizzare tutti i dati relativi:

- agli **studenti** che frequentano i **moduli** di approfondimento e/o i periodi di **Asl**;
- ai **docenti** che hanno il compito di tutor nei periodi di **Asl** e/o insegnano nei **moduli** di approfondimento;
- alle **aziende** che offrono all'istituto i periodi di **Asl**.

### Fase 2 - Specifiche per le operazioni

Il sistema deve offrire i seguenti servizi.

1. Ricercare tutte le scelte (Asl e/o moduli di approfondimento) fatte da uno studente.
2. Ricercare gli studenti che hanno chiesto di frequentare un modulo di approfondimento.
3. Calcolare il numero di aziende che hanno richiesto studenti per l'Asl e il loro elenco.
4. Calcolare il numero di studenti, raggruppati per classe, che hanno chiesto di effettuare il periodo di Asl.

### Fase 3 - Specifiche tecnologiche

Il sistema informativo centralizzato deve essere realizzato con un server di database, al quale possono accedere diversi utenti mediante computer, inseriti nell'intranet dell'istituto.

### Fase 4 - Specifiche per il testing

Per il testing del sistema informativo, ipotizzare una serie di dati iniziali (vedi ad esempio i dati nel database *Asl.mdb*) ed eseguire i servizi creati verificando se i risultati ottenuti sono coerenti con lo stato della base di dati.

## Progetto concettuale

### Progetto dello schema statico (modello logico dei dati)

#### Fatti elementari

Sulla base dell'analisi del testo iniziale del problema e delle specifiche, il sistema può essere schematizzato nei seguenti fatti elementari.

- |   |
|---|
| ■ Uno studente può frequentare più periodi di Asl e/o iscriversi in più moduli di approfondimento.                |
| ■ Un'azienda può offrire più periodi di Asl.  |
| ■ Un docente può essere tutor in più periodi di Asl e/o insegnare in più moduli di approfondimento.               |
| ■ Un modulo può avere un solo docente ma può essere frequentato da più studenti.                                  |
| ■ Un periodo di Asl è frequentato da un solo studente, ha un solo docente tutor ed è offerto da una sola azienda. |

#### Tipi di entità e attributi

I tipi di entità caratteristici del sistema sono:

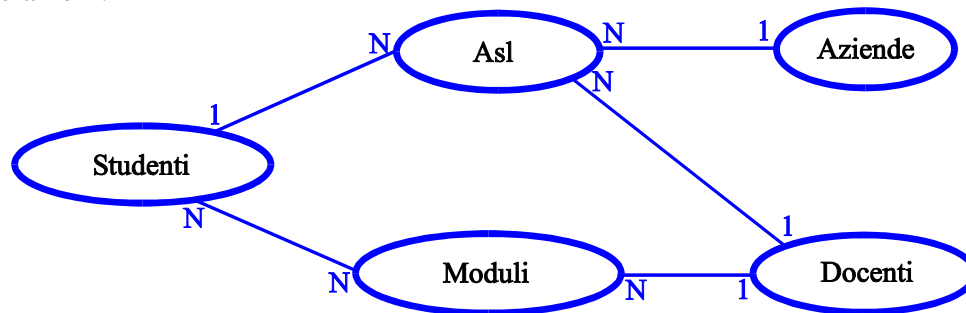
<b>Studenti</b>
<b>Aziende</b>
<b>Docenti</b>
<b>Moduli</b>
<b>Asl</b>

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.

**Documentazione dello schema statico**

**PROGETTO CONCETTUALE:** Asl: Alternanza scuola lavoro;

**INIZIO SCHEMA STATICO:**

**TIPO DI ENTITÀ:** Studenti;

**ATTRIBUTI:**

IDStudente  $\xrightarrow{1 \text{ a } N}$  TIPO: Asl **ATTRIBUTO:** IDStudente;  
 $\xrightarrow{N \text{ a } N}$  TIPO: Moduli **ATTRIBUTO:** IDStudente;

Cognome;  
 Nome;  
 DataNascita;  
 Classe;  
 Sezione;

**REGOLE:**

IDStudente: chiave primaria;  
 Gli attributi IDStudente, Cognome, Nome e DataNascita non possono assumere valori nulli;  
 L'attributo IDStudente ha il formato CS-XXXX (con XXXX cifre numeriche);

**TIPO DI ENTITÀ:** Aziende;

**ATTRIBUTI:**

IDAzienda  $\xrightarrow{1 \text{ a } N}$  TIPO: Asl **ATTRIBUTO:** IDAzienda;

Denominazione;  
 Indirizzo;  
 Contatto;  
 Telefono;  
 Email;

**REGOLE:**

IDAzienda: chiave primaria;  
 IDAzienda, Denominazione, Indirizzo e Telefono, non possono assumere valori nulli;  
 L'attributo IDAzienda ha il formato CA-XXXX (con XXXX cifre numeriche);

**TIPO DI ENTITÀ:** Docenti;

**ATTRIBUTI:**

IDDocente  $\xrightarrow{1 \text{ a } N}$  TIPO: Asl **ATTRIBUTO:** IDDocente;  
 $\xrightarrow{1 \text{ a } N}$  TIPO: Moduli **ATTRIBUTO:** IDDocente;

Cognome;  
 Nome;  
 Materia;

**REGOLE:**

IDDocente: chiave primaria;  
 IDDocente, Cognome e Nome non possono assumere valori nulli;  
 L'attributo IDDocente ha il formato CD-XXXX (con XXXX cifre numeriche);

**TIPO DI ENTITÀ: Moduli;****ATTRIBUTI:**

IDModulo;

Denominazione;

IDDocente  $\xrightarrow{N \rightarrow 1}$  TIPO: Docenti ATTRIBUTO: IDDocente;IDStudente  $\xrightarrow{N \rightarrow N}$  TIPO: Studenti ATTRIBUTO: IDStudente;**REGOLE:**

IDModulo: chiave primaria;

IDModulo, Denominazione, IDDocente e IDStudente non possono assumere valori nulli;

Il tipo di entità Moduli è soggetto al vincolo di integrità referenziale con il tipo Docenti, mediante l'attributo IDDocente (ogni modulo deve sempre fare riferimento al docente che vi insegna);

Il tipo di entità Moduli è soggetto al vincolo di integrità referenziale con il tipo Studenti, mediante l'attributo IDStudente (ogni prenotazione deve sempre fare riferimento ad almeno uno studente registrato nella base di dati);

**TIPO DI ENTITÀ: Asl;****ATTRIBUTI:**

IDAsl;

IDAzienda  $\xrightarrow{N \rightarrow 1}$  TIPO: Aziende ATTRIBUTO: IDAzienda;IDStudente  $\xrightarrow{N \rightarrow 1}$  TIPO: Studenti ATTRIBUTO: IDStudente;IDDocente  $\xrightarrow{N \rightarrow 1}$  TIPO: Docenti ATTRIBUTO: IDDocente;

DataIniziale;

Giorni;

**REGOLE:**

IDAsl: chiave primaria;

IDAsl, IDAzienda, IDStudente e IDDocente non possono assumere valori nulli

L'attributo IDAsl è un codice numerico intero che si incrementa automaticamente di una unità per ogni inserimento di un nuovo tirocinio Asl;

Il tipo di entità Asl è soggetto al vincolo di integrità referenziale con il tipo Aziende, mediante l'attributo IDAzienda (ogni periodo di tirocinio Asl deve sempre fare riferimento a un'azienda);

Il tipo di entità Asl è soggetto al vincolo di integrità referenziale con il tipo Studenti, mediante l'attributo IDStudente (ogni periodo di tirocinio Asl deve sempre fare riferimento a uno studente);

Il tipo di entità Asl è soggetto al vincolo di integrità referenziale con il tipo Docenti, mediante l'attributo IDDocente (ogni periodo di tirocinio Asl deve sempre fare riferimento a un docente tutor);

**FINE SCHEMA STATICO****Progetto dello schema dinamico (le operazioni)****Documentazione**

In base agli obiettivi del problema, la documentazione delle operazioni è la seguente.

**INIZIO SCHEMA DINAMICO:****OPERAZIONE:** aggiornamenti;**DESCRIZIONE:** aggiornamento della base di dati;**TIPO:** Aggiornamento;**UTENTI:** personale istituto responsabile;**ARGOMENTI DI INGRESSO:** i codici delle entità (studenti, docenti, aziende, moduli di approfondimento e periodi di tirocinio Asl) da inserire, modificare o cancellare;**RISULTATI:** inserimento, modifica o cancellazione dei dati;

**OPERAZIONE:** ricerche per studente;

**DESCRIZIONE:** ricerca di tutti i periodi di tirocinio Asl e dei moduli scelti da uno studente;

**TIPO:** Ricerca;

**UTENTI:** personale istituto responsabile;

**ARGOMENTI DI INGRESSO:** Cognome e Nome dello studente;

**RISULTATI:** tutti i dati dei periodi di tirocinio Asl e dei moduli scelti da uno studente;

**OPERAZIONE:** ricerca per modulo;

**DESCRIZIONE:** ricerca gli studenti che hanno chiesto di frequentare un modulo di approfondimento;

**TIPO:** Ricerca, Ordinamento;

**UTENTI:** personale istituto responsabile;

**ARGOMENTI DI INGRESSO:** IDModulo;

**RISULTATI:** Cognome, Nome, Classe e Sezione degli studenti;

**OPERAZIONE:** calcolo numero aziende per Asl;

**DESCRIZIONE:** calcolo del numero di aziende che hanno richiesto studenti per l'Asl e il loro elenco;

**TIPO:** Ricerca, Elaborazione;

**UTENTI:** personale istituto responsabile;

**ARGOMENTI DI INGRESSO:** nessuno;

**RISULTATI:** numero delle aziende e il loro elenco;

**OPERAZIONE:** calcolo numero studenti per classe;

**DESCRIZIONE:** calcolo del numero di studenti, raggruppati per classe, che hanno chiesto di effettuare il periodo di Asl;

**TIPO:** Ricerca, Elaborazione;

**UTENTI:** personale istituto responsabile;

**ARGOMENTI DI INGRESSO:** nessuno;

**RISULTATI:** una tabella in cui per ogni classe è riportato il numero di studenti che ha svolto periodi di Asl;

**FINE SCHEMA DINAMICO**

**FINE PROGETTO.**

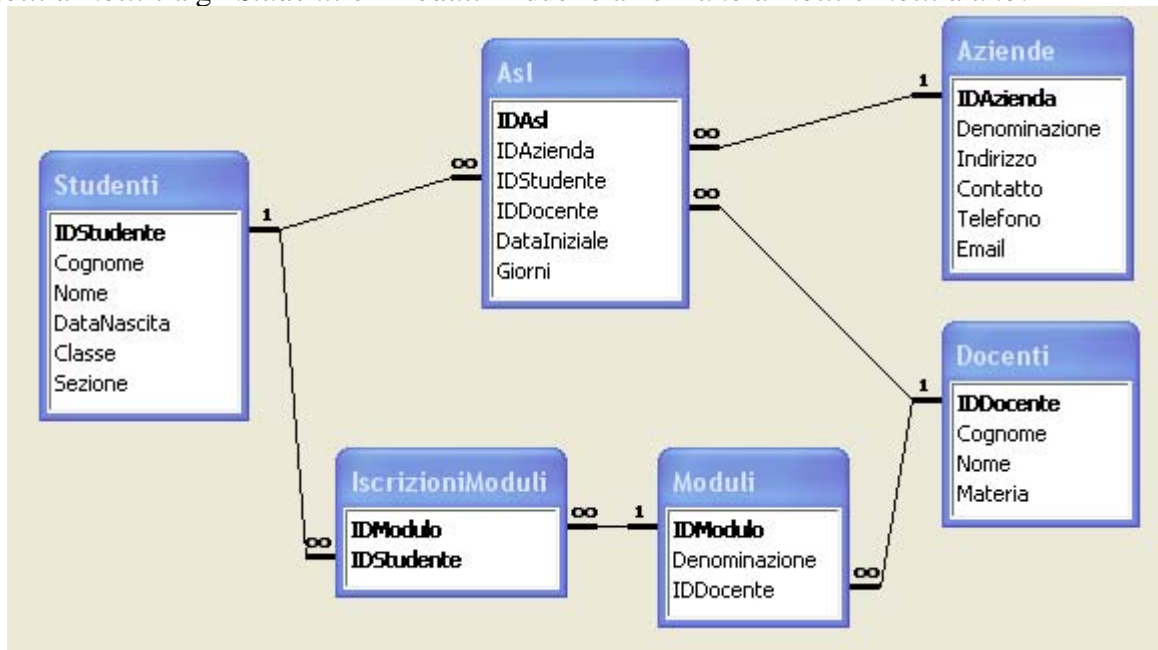
## Realizzazione

### 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.

Lo schema logico relazionale del database per la gestione dei periodi di Asl in un istituto diventa quindi quello riportato nel seguito, che soddisfa le tre forme normali. Rispetto al diagramma E-R, è stata aggiunta la nuova tabella *IscrizioniModuli* per trasformare la relazione *molti a molti* tra gli *Studenti* e i *Moduli* in due relazioni *uno a molti* e *molti a uno*.



### Creazione del database

Nel seguito è riportato il codice sorgente interpretato da un RDBMS (script), scritto nel linguaggio standard SQL, per la creazione dello schema logico del database.

```

CREATE DATABASE Asl

CREATE TABLE Studenti
(
  IDStudente CHAR(7) PRIMARY KEY,
  Cognome CHAR(20) NOT NULL,
  Nome CHAR(20) NOT NULL,
  DataNascita DATE NOT NULL,
  Classe INTEGER,
  Sezione CHAR(5),
  CHECK (IDStudente LIKE 'CS-____')
)

CREATE TABLE Docenti
(
  IDDocente CHAR(7) PRIMARY KEY,
  Cognome CHAR(20) NOT NULL,
  Nome CHAR(20) NOT NULL,
  Materia CHAR(20),
  CHECK (IDDocente LIKE 'CD-____')
)
  
```

```

CREATE TABLE Aziende
(
  IDAzienda      CHAR(7)  PRIMARY KEY,
  Denominazione CHAR(20) NOT NULL,
  Indirizzo      CHAR(30) NOT NULL,
  Contatto       CHAR(30),
  Telefono       CHAR(15) NOT NULL,
  Email          CHAR(30),
  CHECK (IDAzienda LIKE 'CA-____')
)

CREATE TABLE Moduli
(
  IDModulo      CHAR(7)  PRIMARY KEY,
  Denominazione CHAR(20) NOT NULL,
  IDDocente     CHAR(7)  NOT NULL,
  CHECK (IDDocente LIKE 'CM-____'),
  FOREIGN KEY (IDDocente) REFERENCES Docenti(IDDocente)
)

CREATE TABLE IscrizioniModuli
(
  IDModulo      CHAR(7)  NOT NULL,
  IDStudiante   CHAR(7)  NOT NULL,
  PRIMARY KEY (IDModulo, IDStudiante),
  FOREIGN KEY (IDStudiante) REFERENCES Studenti(IDStudiante),
  FOREIGN KEY (IDModulo) REFERENCES Moduli(IDModulo)
)

CREATE TABLE Asl
(
  IDAsl         INTEGER IDENTITY(1,1),
  IDAzienda     CHAR(7)  NOT NULL,
  IDStudiante   CHAR(7)  NOT NULL,
  IDDocente     CHAR(7)  NOT NULL,
  DataIniziale DATE,
  Giorni        INTEGER,
  PRIMARY KEY (IDAsl),
  FOREIGN KEY (IDStudiante) REFERENCES Studenti(IDStudiante),
  FOREIGN KEY (IDAzienda) REFERENCES Aziende(IDAzienda),
  FOREIGN KEY (IDDocente) REFERENCES Docenti(IDDocente)
)


```

Lo script deve essere eseguito (una sola volta) in un RDBMS per la creazione delle tabelle e delle relazioni della base di dati.

### Creazione del database nel RDBMS Access

Lo schema logico 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 6 script SQL, documentati nella tabella che segue. La sintassi standard SQL è stata adattata a quella implementata nel RDBMS Access.

Tabelle di base	Codice SQL nella modalità <small>SQL</small> Visualizzazione SQL
	<p style="text-align: center;">Nome query: ScriptCreaTabellaStudenti</p> <pre> CREATE TABLE Studenti (   IDStudiante  CHAR(7)  PRIMARY KEY,   Cognome      CHAR(20) NOT NULL,   Nome         CHAR(20) NOT NULL,   DataNascita  DATE NOT NULL,   Classe       INTEGER,   Sezione      CHAR(5) ); </pre>

<table border="1"> <thead> <tr> <th colspan="3">Docenti : Tabella</th> </tr> <tr> <th></th> <th>Nome campo</th> <th>Tipo dati</th> </tr> </thead> <tbody> <tr> <td>🔑</td> <td>IDDocente</td> <td>Testo</td> </tr> <tr> <td></td> <td>Cognome</td> <td>Testo</td> </tr> <tr> <td></td> <td>Nome</td> <td>Testo</td> </tr> <tr> <td></td> <td>Materia</td> <td>Testo</td> </tr> </tbody> </table>	Docenti : Tabella				Nome campo	Tipo dati	🔑	IDDocente	Testo		Cognome	Testo		Nome	Testo		Materia	Testo	<p><b>Nome query:</b> ScriptCreaTabellaDocenti</p> <pre>CREATE TABLE Docenti (   IDDocente CHAR(7) PRIMARY KEY,   Cognome CHAR(20) NOT NULL,   Nome CHAR(20) NOT NULL,   Materia CHAR(20) );</pre>						
Docenti : Tabella																									
	Nome campo	Tipo dati																							
🔑	IDDocente	Testo																							
	Cognome	Testo																							
	Nome	Testo																							
	Materia	Testo																							
<b>Tabelle di base</b>		<b>Codice SQL nella modalità <small>\$SQL</small> Visualizzazione SQL</b>																							
<table border="1"> <thead> <tr> <th colspan="3">Aziende : Tabella</th> </tr> <tr> <th></th> <th>Nome campo</th> <th>Tipo dati</th> </tr> </thead> <tbody> <tr> <td>🔑</td> <td>IDAzienda</td> <td>Testo</td> </tr> <tr> <td></td> <td>Denominazione</td> <td>Testo</td> </tr> <tr> <td></td> <td>Indirizzo</td> <td>Testo</td> </tr> <tr> <td></td> <td>Contatto</td> <td>Testo</td> </tr> <tr> <td></td> <td>Telefono</td> <td>Testo</td> </tr> <tr> <td></td> <td>Email</td> <td>Testo</td> </tr> </tbody> </table>	Aziende : Tabella				Nome campo	Tipo dati	🔑	IDAzienda	Testo		Denominazione	Testo		Indirizzo	Testo		Contatto	Testo		Telefono	Testo		Email	Testo	<p><b>Nome query:</b> ScriptCreaTabellaAziende</p> <pre>CREATE TABLE Aziende (   IDAzienda CHAR(7) PRIMARY KEY,   Denominazione CHAR(20) NOT NULL,   Indirizzo CHAR(30) NOT NULL,   Contatto CHAR(30),   Telefono CHAR(15) NOT NULL,   Email CHAR(30) );</pre>
Aziende : Tabella																									
	Nome campo	Tipo dati																							
🔑	IDAzienda	Testo																							
	Denominazione	Testo																							
	Indirizzo	Testo																							
	Contatto	Testo																							
	Telefono	Testo																							
	Email	Testo																							
<table border="1"> <thead> <tr> <th colspan="3">Moduli : Tabella</th> </tr> <tr> <th></th> <th>Nome campo</th> <th>Tipo dati</th> </tr> </thead> <tbody> <tr> <td>🔑</td> <td>IDModulo</td> <td>Testo</td> </tr> <tr> <td></td> <td>Denominazione</td> <td>Testo</td> </tr> <tr> <td></td> <td>IDDocente</td> <td>Testo</td> </tr> </tbody> </table>	Moduli : Tabella				Nome campo	Tipo dati	🔑	IDModulo	Testo		Denominazione	Testo		IDDocente	Testo	<p><b>Nome query:</b> ScriptCreaTabellaModuli</p> <pre>CREATE TABLE Moduli (   IDModulo CHAR(7) PRIMARY KEY,   Denominazione CHAR(20) NOT NULL,   IDDocente CHAR(7) NOT NULL,   FOREIGN KEY (IDDocente)     REFERENCES Docenti(IDDocente) );</pre>									
Moduli : Tabella																									
	Nome campo	Tipo dati																							
🔑	IDModulo	Testo																							
	Denominazione	Testo																							
	IDDocente	Testo																							
<table border="1"> <thead> <tr> <th colspan="3">IscrizioniModuli : Tabella</th> </tr> <tr> <th></th> <th>Nome campo</th> <th>Tipo dati</th> </tr> </thead> <tbody> <tr> <td>🔑</td> <td>IDModulo</td> <td>Testo</td> </tr> <tr> <td>🔑</td> <td>IDStudente</td> <td>Testo</td> </tr> </tbody> </table>	IscrizioniModuli : Tabella				Nome campo	Tipo dati	🔑	IDModulo	Testo	🔑	IDStudente	Testo	<p><b>Nome query:</b> ScriptCreaTabellaIscrizioniModuli</p> <pre>CREATE TABLE IscrizioniModuli (   IDModulo CHAR(7) NOT NULL,   IDStudente CHAR(7) NOT NULL,   PRIMARY KEY (IDModulo, IDStudente),   FOREIGN KEY (IDStudente)     REFERENCES Studenti(IDStudente),   FOREIGN KEY (IDModulo)     REFERENCES Moduli(IDModulo) );</pre>												
IscrizioniModuli : Tabella																									
	Nome campo	Tipo dati																							
🔑	IDModulo	Testo																							
🔑	IDStudente	Testo																							
<table border="1"> <thead> <tr> <th colspan="3">Asl : Tabella</th> </tr> <tr> <th></th> <th>Nome campo</th> <th>Tipo dati</th> </tr> </thead> <tbody> <tr> <td>🔑</td> <td>IDAsl</td> <td>Contatore</td> </tr> <tr> <td></td> <td>IDAzienda</td> <td>Testo</td> </tr> <tr> <td></td> <td>IDStudente</td> <td>Testo</td> </tr> <tr> <td></td> <td>IDDocente</td> <td>Testo</td> </tr> <tr> <td></td> <td>DataIniziale</td> <td>Data/ora</td> </tr> <tr> <td></td> <td>Giorni</td> <td>Numerico</td> </tr> </tbody> </table>	Asl : Tabella				Nome campo	Tipo dati	🔑	IDAsl	Contatore		IDAzienda	Testo		IDStudente	Testo		IDDocente	Testo		DataIniziale	Data/ora		Giorni	Numerico	<p><b>Nome query:</b> ScriptCreaTabellaAsl</p> <pre>CREATE TABLE Asl (   IDAsl COUNTER PRIMARY KEY,   IDAzienda CHAR(7) NOT NULL,   IDStudente CHAR(7) NOT NULL,   IDDocente CHAR(7) NOT NULL,   DataIniziale DATE,   Giorni INTEGER,   FOREIGN KEY (IDStudente)     REFERENCES Studenti(IDStudente),   FOREIGN KEY (IDAzienda)     REFERENCES Aziende(IDAzienda),   FOREIGN KEY (IDDocente)     REFERENCES Docenti(IDDocente) );</pre>
Asl : Tabella																									
	Nome campo	Tipo dati																							
🔑	IDAsl	Contatore																							
	IDAzienda	Testo																							
	IDStudente	Testo																							
	IDDocente	Testo																							
	DataIniziale	Data/ora																							
	Giorni	Numerico																							

Gli schemi delle tabelle di base del database sono stati poi completati in modo interattivo (nel file di database *Asl.mdb*) definendo i vincoli di integrità dei dati e quelli referenziali sulle relazioni.

## Realizzazione dei servizi

I servizi del sistema informativo sono stati realizzati in Access per cui verrà presentata l'interrogazione scritta nell'implementazione SQL di questo RDBMS (che presenta alcune differenze di sintassi rispetto allo standard SQL). Le interrogazioni SQL sono state collaudate digitando direttamente il codice in un oggetto *Query* di Access nella modalità *Visualizzazione SQL*.

### Servizio: ricerche per studente

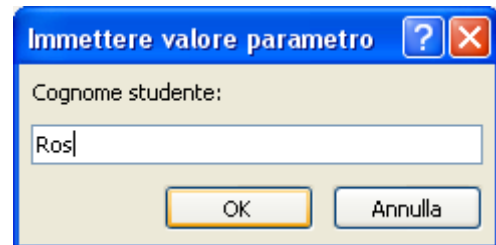
Questa operazione deve essere realizzata con due query di cui la prima (*RicercaModuliStudente*) ricerca tutti i dati dei periodi di tirocinio, mentre la seconda (*RicercaAslStudente*) fornisce le righe relative ai periodi di tirocinio Asl.

Nome query: *RicercaModuliStudente*

#### Codice sorgente SQL

```
SELECT Moduli.*
FROM Studenti, IscrizioniModuli, Moduli
WHERE Studenti.IDStudente = IscrizioniModuli.IDStudente AND
      IscrizioniModuli.IDModulo = Moduli.IDModulo AND
      Studenti.Cognome=[Cognome studente:] AND Studenti.Nome=[Nome studente:]
```

Nel codice precedente, così come in quelli dei servizi successivi, i nomi racchiusi tra parentesi quadre (ad esempio [*Cognome studente:*]) rappresentano parametri di ingresso il cui valore viene richiesto da Access prima dell'esecuzione dell'operazione con una finestra di dialogo del tipo descritta nella figura a fianco.



Nome query: *RicercaAslStudente*

#### Codice sorgente SQL

```
SELECT Asl.*
FROM Studenti INNER JOIN Asl ON Studenti.IDStudente = Asl.IDStudente
WHERE Studenti.Cognome = [Cognome studente] AND Studenti.Nome = [Nome studente]
```

### Servizio: ricerca per modulo

Nome query: *RicercaPerModulo*

#### Codice sorgente SQL

```
SELECT Studenti.Cognome, Studenti.Nome, Studenti.Classe, Studenti.Sezione
FROM Studenti, IscrizioniModuli, Moduli
WHERE Studenti.IDStudente = IscrizioniModuli.IDStudente AND
      IscrizioniModuli.IDModulo = Moduli.IDModulo AND
      Moduli.IDModulo= [Codice del modulo:]
ORDER BY Studenti.Cognome, Studenti.Nome
```

**Servizio: calcolo numero aziende per Asl**

Questa operazione può essere realizzata con due query di cui la prima fornisce (*RicercaAziendeAsl*) l'elenco delle aziende che hanno richiesto studenti per l'Asl (solo le aziende che hanno delle registrazioni nella tabella *Asl*), mentre la seconda (*NumeroAziendeAsl*) ne calcola il loro numero mediante la tabella derivata (vista) *RicercaAziendeAsl*.

**Nome query:** *RicercaAziendeAsl*

**Codice sorgente SQL**

```
SELECT DISTINCT Aziende.*
FROM Aziende INNER JOIN Asl ON Aziende.IDAzienda = Asl.IDAzienda
```

**Nome query:** *NumeroAziendeAsl*

**Codice sorgente SQL**

```
SELECT COUNT(Aziende.IDAzienda) AS "Numero aziende che ha offerto Asl"
FROM RicercaAziendeAsl
```

**Servizio: calcolo numero studenti per classe**

**Nome query:** *NumeroStudentiAslPerClasse*

**Codice sorgente SQL**

```
SELECT COUNT(Studenti.IDStudente) As "Numero studenti",
       Studenti.Classe, Studenti.Sezione
FROM Studenti INNER JOIN Asl ON Studenti.IDStudente = Asl.IDStudente
GROUP BY Studenti.Classe, Studenti.Sezione
```