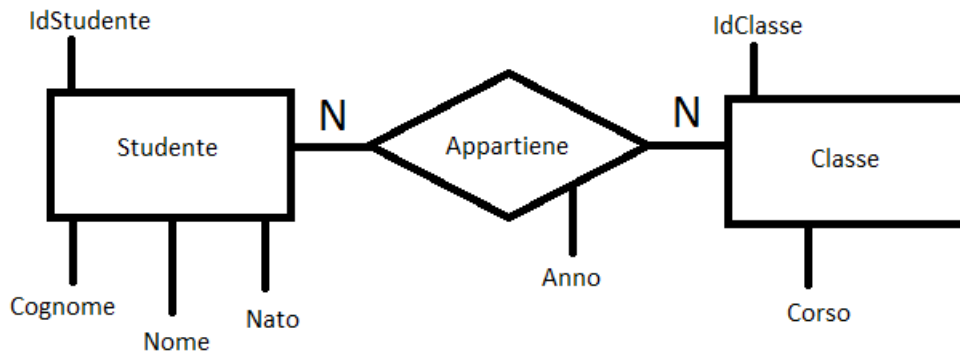


## Esercizio con connessione a database remoto<sup>1</sup>

Si progetta un DB per archiviare informazioni relative al *percorso scolastico* di studenti frequentanti classi di un dato Istituto di Istruzione Superiore. Per semplicità, si considera una sola specializzazione (secondo biennio e classe quinta).

### Modello E-R



Ogni studente, in anni diversi, frequenta (appartiene a) più classi  
Ogni classe è frequentata da più studenti

### Modello logico

Classi (IdClasse, Corso)

Studenti (IdStudente, Cognome, Nome, Nato)

Appartiene (IdStud, IdCla, Anno)

### Modello fisico

```
CREATE TABLE Studenti
( IdStudente int IDENTITY(1,1) PRIMARY KEY,
  Cognome varchar(255) NOT NULL,
  Nome varchar(255) NOT NULL,
  Nato date
);
```

```
CREATE TABLE Classi
( IdClasse int IDENTITY(1,1) PRIMARY KEY,
  Corso varchar(255) NOT NULL
);
```

```
CREATE TABLE Appartiene
( IdStud int NOT NULL FOREIGN KEY REFERENCES Studenti(IdStudente),
  IdCla int NOT NULL FOREIGN KEY REFERENCES Classi(IdClasse),
  Anno varchar(255) NOT NULL
);
```

<sup>1</sup> Con account [somee.com](https://somee.com) uso di DBMS SQL Server e tecnologia ASP.Net

## Popolate le tabelle:

```
INSERT INTO Classi (Corso)
VALUES ('5AI' );
INSERT INTO Classi (Corso)
VALUES ('4AI');
INSERT INTO Classi (Corso)
VALUES ('3AI');
INSERT INTO Classi (Corso)
VALUES ('5BI');
INSERT INTO Classi (Corso)
VALUES ('4BI');
INSERT INTO Classi (Corso)
VALUES ('3BI');
```

### SQL Query Result

IdClasse	Corso
1	5AI
2	4AI
3	3AI
4	5BI
5	4BI
6	3BI

```
INSERT INTO Appartiene (IdStud,IdCla, Anno) VALUES (1, 1, '2020-21');
INSERT INTO Appartiene (IdStud,IdCla, Anno) VALUES (1, 2, '2019-20');
INSERT INTO Appartiene (IdStud,IdCla, Anno) VALUES (1, 3, '2018-19');
INSERT INTO Appartiene (IdStud,IdCla, Anno) VALUES (2, 4,'2020-21' );
INSERT INTO Appartiene (IdStud,IdCla, Anno) VALUES (2, 5, '2019-20');
INSERT INTO Appartiene (IdStud,IdCla, Anno) VALUES (2, 6, '2018-19');
INSERT INTO Appartiene (IdStud,IdCla, Anno) VALUES (3, 3, '2019-20' );
INSERT INTO Appartiene (IdStud,IdCla, Anno) VALUES (3, 2, '2020-21');
INSERT INTO Appartiene (IdStud,IdCla, Anno) VALUES (4, 3, '2020-21');
INSERT INTO Appartiene (IdStud,IdCla, Anno) VALUES (5, 3, '2019-20');
INSERT INTO Appartiene (IdStud,IdCla, Anno) VALUES (5, 6,'2020-21' );
```

### SQL Query Result

IdStud	IdCla	Anno
1	1	2020-21
1	2	2019-20
1	3	2018-19
2	4	2020-21
2	5	2019-20
2	6	2018-19
3	3	2019-20
3	2	2020-21
4	3	2020-21
5	3	2019-20
5	6	2020-21

2 studenti (Andri e Bogghi sempre promossi) dalla 3 alla 5 nel 2020-21  
uno corso AI l'altro corso BI

sorelle Allegri: Dora frequentante 3AI nel 2020-21

e Sara un anno prima, promossa, frequenta la 4AI nell'anno 2020-21

Uno studente bocciato in 3AI (Petri Luca), ripete in 3BI.

## Si realizzano query di estrazione

### Tutte le informazioni ordinando per Corso

```
SELECT Studenti.Cognome, Studenti.Nome,
       Year(Studenti.Nato) AS Nato, Classi.Corso,
       Appartiene.Anno
FROM Studenti, Classi, Appartiene
WHERE Classi.IdClasse = Appartiene.IdCla
AND Studenti.IdStudente = Appartiene.IdStud
Order By Corso
```

Cognome	Nome	Nato	Corso	Anno
Andri	Paolo	2003	3AI	2018-19
Allegri	Sara	2004	3AI	2019-20
Allegri	Dora	2005	3AI	2020-21
Petri	Luca	2004	3AI	2019-20
Bolghi	Mauro	2002	3BI	2018-19
Petri	Luca	2004	3BI	2020-21
Andri	Paolo	2003	4AI	2019-20
Allegri	Sara	2004	4AI	2020-21
Bolghi	Mauro	2002	4BI	2019-20
Andri	Paolo	2003	5AI	2020-21
Bolghi	Mauro	2002	5BI	2020-21

*Tutte le informazioni ordinando per Anno scolastico*

```
SELECT Studenti.Cognome, Studenti.Nome,  
       Year(Studenti.Nato) AS Nato, Classi.Corso,  
       Appartiene.Anno  
FROM Studenti, Classi, Appartiene  
WHERE Classi.IdClasse = Appartiene.IdCla  
AND Studenti.IdStudente = Appartiene.IdStud  
Order By Anno
```

Cognome	Nome	Nato	Corso	Anno
Andri	Paolo	2003	3AI	2018-19
Bolghi	Mauro	2002	3BI	2018-19
Allegri	Sara	2004	3AI	2019-20
Andri	Paolo	2003	4AI	2019-20
Petri	Luca	2004	3AI	2019-20
Bolghi	Mauro	2002	4BI	2019-20
Petri	Luca	2004	3BI	2020-21
Andri	Paolo	2003	5AI	2020-21
Bolghi	Mauro	2002	5BI	2020-21
Allegri	Sara	2004	4AI	2020-21
Allegri	Dora	2005	3AI	2020-21

*Tutte le informazioni ordinando per Nominativo dello studente*

```
SELECT Studenti.Cognome, Studenti.Nome,  
       Year(Studenti.Nato) AS Nato, Classi.Corso,  
       Appartiene.Anno  
FROM Studenti, Classi, Appartiene  
WHERE Classi.IdClasse = Appartiene.IdCla  
AND Studenti.IdStudente = Appartiene.IdStud  
Order By Cognome, Nome
```

Cognome	Nome	Nato	Corso	Anno
Allegri	Dora	2005	3AI	2020-21
Allegri	Sara	2004	3AI	2019-20
Allegri	Sara	2004	4AI	2020-21
Andri	Paolo	2003	5AI	2020-21
Andri	Paolo	2003	4AI	2019-20
Andri	Paolo	2003	3AI	2018-19
Bolghi	Mauro	2002	5BI	2020-21
Bolghi	Mauro	2002	4BI	2019-20
Bolghi	Mauro	2002	3BI	2018-19
Petri	Luca	2004	3AI	2019-20
Petri	Luca	2004	3BI	2020-21

*Le informazioni di studenti frequentanti  
in dato Anno scolastico  
ordinando per Nominativo dello studente*

```
SELECT Studenti.Cognome, Studenti.Nome,  
       Year(Studenti.Nato) AS Nato, Classi.Corso,  
       Appartiene.Anno  
FROM Studenti, Classi, Appartiene  
WHERE Classi.IdClasse = Appartiene.IdCla  
AND Studenti.IdStudente = Appartiene.IdStud  
AND Anno= '2020-21'  
Order By Cognome, Nome
```

SQL Query Result

Cognome	Nome	Nato	Corso	Anno
Allegri	Dora	2005	3AI	2020-21
Allegri	Sara	2004	4AI	2020-21
Andri	Paolo	2003	5AI	2020-21
Bolghi	Mauro	2002	5BI	2020-21
Petri	Luca	2004	3BI	2020-21

**Le informazioni di studenti frequentanti un dato Corso ordinando per Nominativo dello studente**

```
SELECT Studenti.Cognome, Studenti.Nome,
       Year(Studenti.Nato) AS Nato, Classi.Corso,
       Appartiene.Anno
FROM Studenti, Classi, Appartiene
WHERE Classi.IdClasse = Appartiene.IdCla
AND Studenti.IdStudente = Appartiene.IdStud
AND Corso Like '%AI'
Order By Cognome, Nome
```

Cognome	Nome	Nato	Corso	Anno
Allegri	Dora	2005	3AI	2020-21
Allegri	Sara	2004	3AI	2019-20
Allegri	Sara	2004	4AI	2020-21
Andri	Paolo	2003	5AI	2020-21
Andri	Paolo	2003	4AI	2019-20
Andri	Paolo	2003	3AI	2018-19
Petri	Luca	2004	3AI	2019-20

```
SELECT Studenti.Cognome, Studenti.Nome,
       Year(Studenti.Nato) AS Nato, Classi.Corso,
       Appartiene.Anno
FROM Studenti, Classi, Appartiene
WHERE Classi.IdClasse = Appartiene.IdCla
AND Studenti.IdStudente = Appartiene.IdStud
AND Corso Like '%BI'
Order By Cognome, Nome
```

Cognome	Nome	Nato	Corso	Anno
Bolghi	Mauro	2002	5BI	2020-21
Bolghi	Mauro	2002	4BI	2019-20
Bolghi	Mauro	2002	3BI	2018-19
Petri	Luca	2004	3BI	2020-21

**Le informazioni sul percorso di un dato studente**

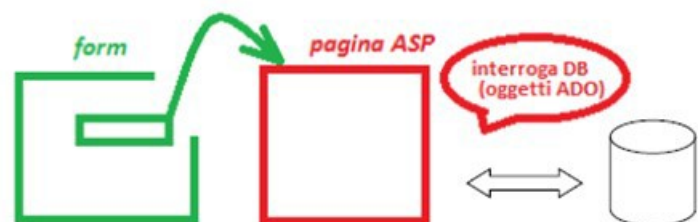
```
SELECT Studenti.Cognome, Studenti.Nome,
       Year(Studenti.Nato) AS Nato, Classi.Corso,
       Appartiene.Anno
FROM Studenti, Classi, Appartiene
WHERE Classi.IdClasse = Appartiene.IdCla
AND Studenti.IdStudente = Appartiene.IdStud
AND Cognome='Bolghi'
AND Nome='Mauro'
```

Cognome	Nome	Nato	Corso	Anno
Bolghi	Mauro	2002	5BI	2020-21
Bolghi	Mauro	2002	4BI	2019-20
Bolghi	Mauro	2002	3BI	2018-19

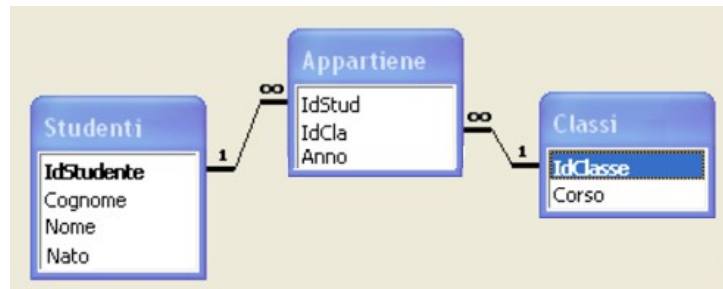
**Consultazione da remoto**

Cognome	Nome	Corso	Anno
Andri	Paolo	3AI	2018-19
Allegri	Sara	3AI	2019-20
Allegri	Dora	3AI	2020-21
Petri	Luca	3AI	2019-20
Bolghi	Mauro	3BI	2018-19
Petri	Luca	3BI	2020-21
Andri	Paolo	4AI	2019-20
Allegri	Sara	4AI	2020-21
Bolghi	Mauro	4BI	2019-20
Andri	Paolo	5AI	2020-21
Bolghi	Mauro	5BI	2020-21

Presentazione dei risultati in forma tabellare con effetto come in figura integrando linguaggio HTML e CSS nella [pagina ASP](#) che consente la **connessione al DB remoto**



Per realizzare una query di estrazione di dati da più tabelle (Classi, Studenti, Appartiene) memorizzate in un **database** archiviato su server **remoto** e **visualizzarli sulla finestra del browser**



#### pagina ASP

<%

' Per rendere obbligatorio l'uso di Dim cioè esplicitare le variabili in uso  
Option Explicit

Dim oConn, Rs, sSQL

' Si crea un oggetto sul Server per la connessione a database con strumenti ADO  
Set oConn = Server.CreateObject("ADODB.Connection")

' Si crea un oggetto sul Server per individuare i vari records con strumenti ADO  
Set Rs = Server.CreateObject("ADODB.RecordSet")

' si usa driver opportuno con indipendenza dalla versione del database  
oConn.open ("Driver={SQL Server}; Server=DBMS5BI.mssql.somee.com; Database=DBMS5BI;  
Uid=infcol5ai\_SQLLogin\_1; Pwd=password;")

' Si imposta la sintassi della query di estrazione dati con linguaggio SQL<sup>2</sup>

sSQL ="SELECT Studenti.Cognome, Studenti.Nome, Year(Studenti.Nato) AS Nato, Classi.Corso,  
Appartiene.Anno FROM Studenti, Classi, Appartiene WHERE Classi.IdClasse = Appartiene.IdCla AND  
Studenti.IdStudente = Appartiene.IdStud **Order By Corso**"

' Si deve eseguire la query con **recordset aperto** e connessione stabilita  
Rs.Open sSQL,oConn

' Si visualizza su finestra del browser il risultato – anche parziale – della ricerca:

' ciclo fino a quando il set di record non raggiunge la fine della *tabella logica*

```
Response.write("<table style='background-color: lightgreen'><tr><th>Cognome</th><th>Nome</th><th>Corso</th><th>Anno</th></tr>")
```

```
while not RS.eof
```

```
    Response.write("<tr><td>" & RS("Cognome") & "<td>" & RS("Nome") & "<td>" & RS("Corso") & "<td>" & RS("Anno") & "</tr>")
```

```
    RS.movenext
```

```
wend
```

```
Response.write("</table>")
```

'Si chiude la connessione  
oConn.close

'Si rilasciano le risorse liberando la memoria allocata (cancellazione connessione e recordset)  
Set Rs = Nothing  
Set oConn = Nothing

%>

Richiedendo l'esecuzione della **pagina in modo diretto** o **tramite scheda (form)**

<sup>2</sup> Structured Query Language. Per la sintassi vedi <http://www.w3schools.com/sql/default.asp>

Per connettersi alla sorgente di dati si sono creati un oggetto **connessione** ed un oggetto **recordset**.

Quando la connessione è stata stabilita, si genera una **tabella dinamica** ed è necessario che il recordset si mantenga aperto (nomeRecordSet.open sSQL, oConn).

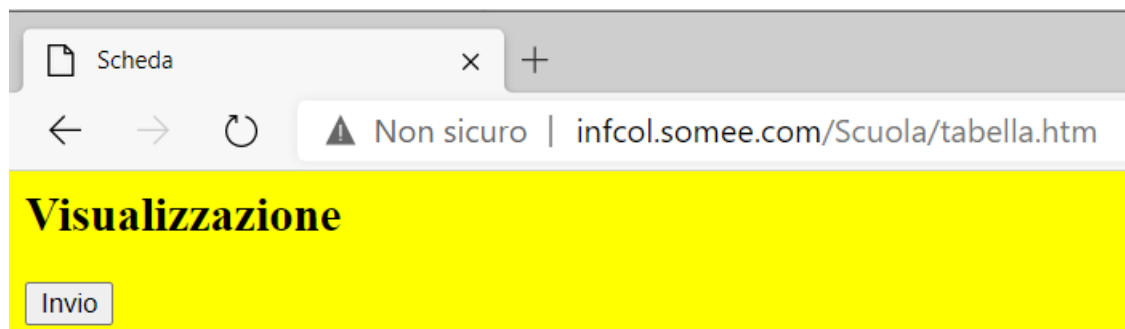
Esiste una seconda tecnica che utilizza l'esecuzione<sup>3</sup> della query (tipicamente updates, inserts e deletes) con sintassi:

nomeConnessioneDb.execute (sintassiSQL)

anche senza invocare esplicitamente il metodo di creazione dell'oggetto sul Server che permetterà di individuare i vari records con strumenti ADO.

Set RS = oConn.execute (sSQL)

Si può richiedere l'esecuzione, lanciando la **scheda** <http://infcoul.somee.com/Scuola/tabella.htm>



```
<form method="get" action="http://infcoul.somee.com/scuola/tabella.asp">  
  <p><input type = "submit" value = "invio"></p>  
</form>
```

oppure lanciando direttamente <http://infcoul.somee.com/Scuola/tabella.asp>

A screenshot of a web browser window showing a table of student data. The address bar shows 'infcoul.somee.com/Scuola/tabella.asp' with a warning icon and the text 'Non sicuro'. The table has four columns: 'Cognome', 'Nome', 'Corso', and 'Anno'. The data is as follows:

Cognome	Nome	Corso	Anno
Andri	Paolo	3AI	2018-19
Allegri	Sara	3AI	2019-20
Allegri	Dora	3AI	2020-21
Petri	Luca	3AI	2019-20
Bolghi	Mauro	3BI	2018-19
Petri	Luca	3BI	2020-21
Andri	Paolo	4AI	2019-20
Allegri	Sara	4AI	2020-21
Bolghi	Mauro	4BI	2019-20
Andri	Paolo	5AI	2020-21
Bolghi	Mauro	5BI	2020-21

At the bottom right of the page, there is a link 'Web hosting by Somee.com'.

<sup>3</sup> Il metodo **execute** dell'oggetto Connection recupera i dati dal DB e li inserisce nell'oggetto Recordset