

CORSO DI PROGRAMMAZIONE

GESTIONE DATABASE LOCALE

16-01_DatabaseLocaleSql[ver_1]



Questa dispensa è rilasciata sotto la licenza Creative Common CC BY-NC-SA. Chiunque può copiare, distribuire, modificare, creare opere derivate dall'originale, ma non a scopi commerciali, a condizione che venga riconosciuta la paternità dell'opera all'autore e che alla nuova opera vengano attribuite le stesse licenze dell'originale.

Versione del: **18/01/2019**
Revisione **2**
numero:

Proff. M. Cherchi, A. Zoccheddu

**DIPARTIMENTO
INFORMATICA E TELECOMUNICAZIONI**





DATABASE LOCALE

CREAZIONE DI UN DATABASE LOCALE CON VISUAL STUDIO

È possibile esplorare le attività di base, ad esempio l'aggiunta di tabelle e la definizione di colonne, tramite Visual Studio per creare e aggiornare un file di database locale in LocalDB di SQL Server Express. Dopo aver completato questa procedura, sarà possibile individuare funzionalità più avanzate utilizzando il database locale come punto iniziale per le altre procedure che la richiedono.

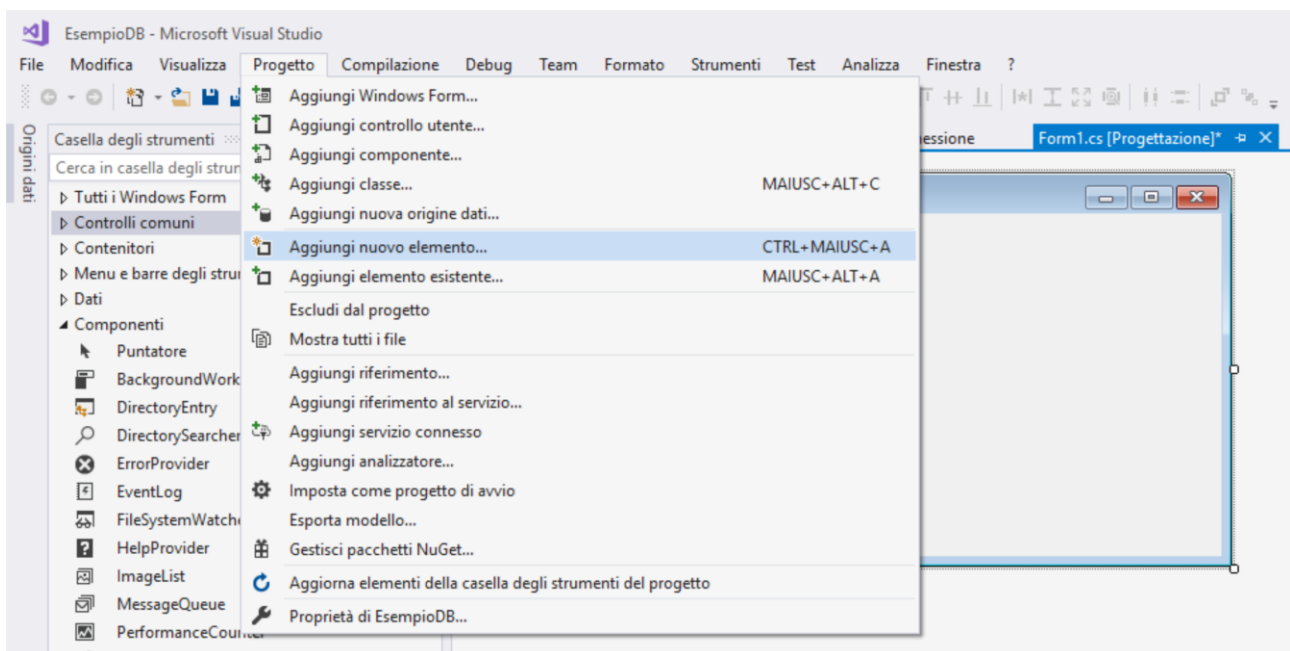
Durante questa procedura, verranno illustrate le seguenti attività:

- Creazione di un progetto e di un file di database locale.
- Creazione di tabelle, colonne, chiavi primarie ed esterne.
- Inserimento di dati nelle tabelle.

CREAZIONE DI UN PROGETTO E DI UN FILE DI DATABASE LOCALE

Per creare un progetto e un file di database

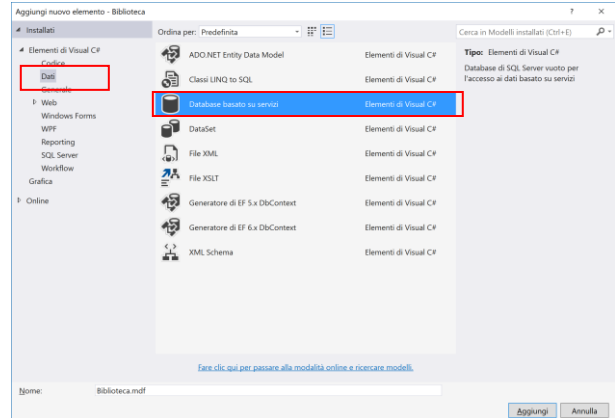
1. Creare un progetto Windows Form denominato DatabaseEsempio.
2. Sulla barra dei menu scegliere **Progetto, Aggiungi nuovo elemento**.



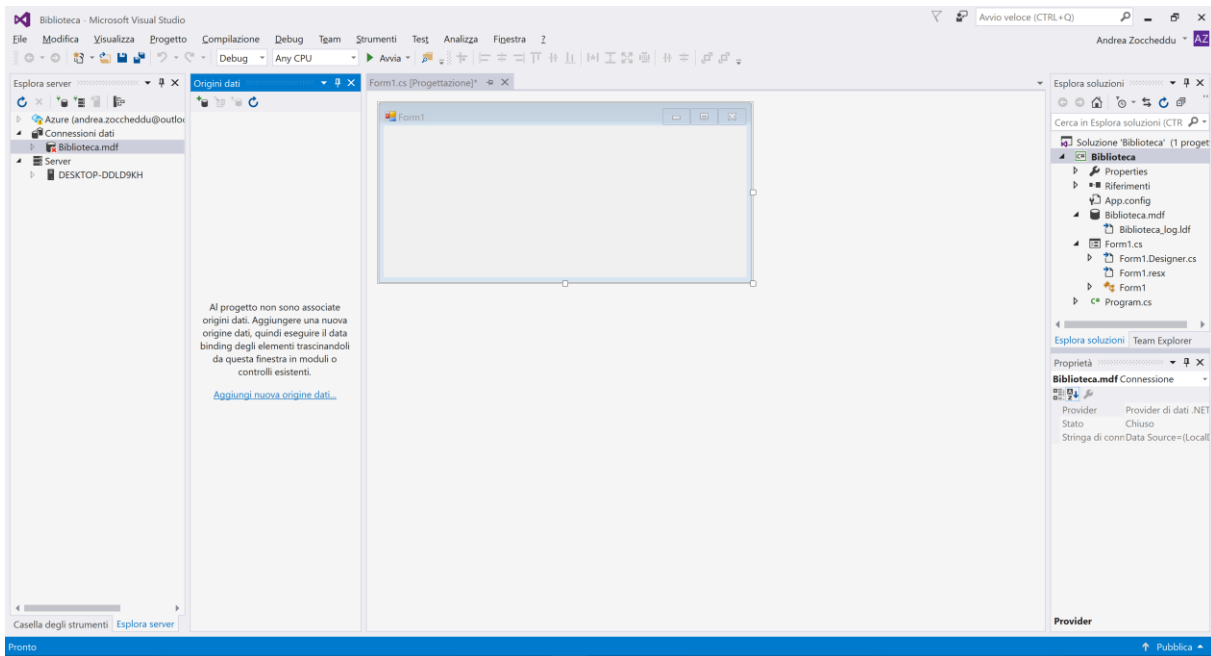
Verrà visualizzata la finestra di dialogo **Aggiungi nuovo elemento** per consentire l'aggiunta di elementi appropriati in un progetto Windows Form.



3. Selezionare la voce Dati di Visual C# sulla sinistra
4. Nell'elenco di modelli di elemento, scorrere fino a visualizzare Database basato su servizi, quindi selezionarlo.
5. Denominare il database Biblioteca, quindi selezionare il pulsante Aggiungi.



6. Se la finestra **Origini dati** non è aperta, scegliere la combinazione di tasti MAIUSC-ALT-D oppure sulla barra dei menu scegliere **Visualizza, Altre finestre, Origini dati** per aprirla.



7. Nella finestra Origini dati scegliere il collegamento **Aggiungi nuova origine dati**.
8. In **Configurazione guidata origine dati** scegliere **Avanti** quattro volte per accettare le impostazioni predefinite, quindi scegliere **Fine**.

Se si apre la finestra delle proprietà per il database, è possibile visualizzare la stringa di connessione relativa e il percorso del file primario con estensione mdf.



Configurazione guidata origine dati

Selezione un tipo di origine dati

Specificare l'origine dei dati usati dall'applicazione:

Database Servizio Oggetto SharePoint

Consente di connettersi a un database e scegliere gli oggetti di database per l'applicazione.

< Indietro Avanti > Fine Annulla

Configurazione guidata origine dati

Scegli modello di database

Specificare il tipo di modello di database da utilizzare.

Dataset

Il modello di database scelto determina i tipi di oggetti dati utilizzati dal codice dell'applicazione. Al progetto verrà aggiunto un file del set di dati.

< Indietro Avanti > Fine Annulla

Configurazione guidata origine dati

Selezione connessione dati

Specificare la connessione dati usata dall'applicazione per connettersi al database.

Biblioteca.mdf Nuova connessione...

La stringa di connessione sembra contenere dati sensibili, ad esempio una password, richiesti per connettersi al database. L'archiviazione dei dati sensibili nella stringa di connessione può generare problemi di sicurezza. Includere i dati sensibili nella stringa di connessione?

No, escludi i dati sensibili dalla stringa di connessione. Queste informazioni verranno impostate nel codice dell'applicazione.

Sì, includi i dati sensibili nella stringa di connessione.

Stringa di connessione da salvare nell'applicazione. Espandere per visualizzare i dettagli

```
Data Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=[DataDirectory]Biblioteca.mdf;Integrated Security=True
```

< Indietro Avanti > Fine Annulla

Configurazione guidata origine dati

Salva stringa di connessione nel file di configurazione dell'applicazione

L'archiviazione delle stringhe di connessione nel file di configurazione dell'applicazione facilita la manutenzione e la distribuzione. Per salvare la stringa di connessione nel file di configurazione dell'applicazione, immettere un nome nella casella e scegliere Avanti.

Salvare la stringa di connessione nel file di configurazione dell'applicazione?

Sì, salva la connessione come:

BibliotecaConnectionString

< Indietro Avanti > Fine Annulla

Configurazione guidata origine dati

Selezione oggetti di database

Specificare gli oggetti di database da inserire nel DataSet:

Il database selezionato è nuovo o non contiene oggetti. Scegliere Fine per creare un DataSet vuoto. Dopo avere aggiunto oggetti di database al database, aprire il DataSet vuoto in Progettazione DataSet per riconfigurarlo.

Nome DataSet:
BibliotecaDataSet

< Indietro Avanti > Fine Annulla

Esplora server

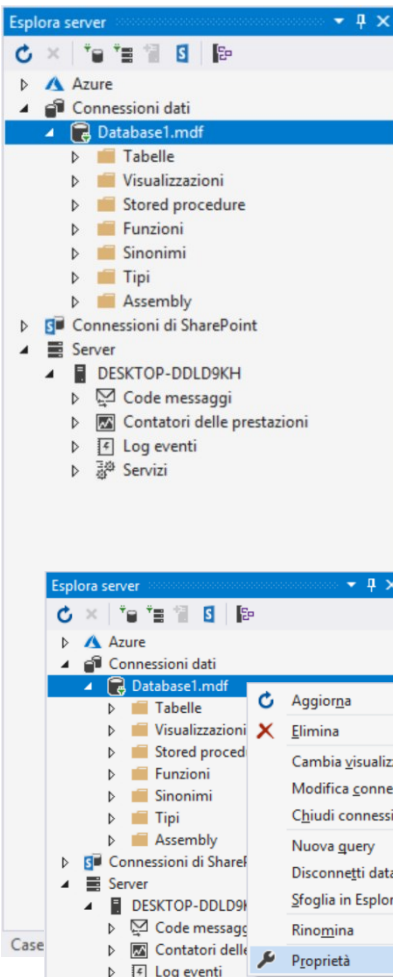
- Azure (andrea.zoccheddu@outlook.com)
- Connessioni dati
 - Biblioteca.mdf
 - Tabelle
 - Visualizzazioni
 - Stored procedure
 - Funzioni
 - Sinonimi
 - Tipi
 - Assembly
- Server
 - DESKTOP-DDLD9KH

Origini dati

- BibliotecaDataSet

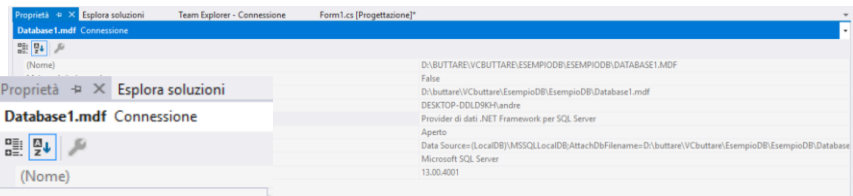
Form1.cs [Progettazione]*

Form1



9. In Visual Studio Express scegliere **Visualizza, Altre finestre, Esplora database** se la finestra non è già aperta. Espandere il nodo **Connessioni dati**, aprire il menu di scelta rapida per il file **Database1.mdf**, quindi scegliere **Proprietà** per aprire la finestra delle proprietà.

10. In altre versioni di Visual Studio scegliere **Visualizza, Esplora server** se la finestra non è già aperta. Espandere il nodo **Connessioni dati**, aprire il menu di scelta rapida per il file **Database1.mdf**, quindi scegliere **Proprietà** per aprire la finestra delle proprietà.

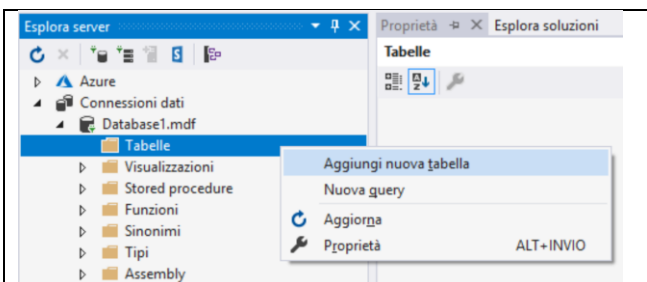


CREAZIONE DI TABELLE, COLONNE, CHIAVI PRIMARIE ED ESTERNE

In questa sezione verranno create un paio di tabelle, una chiave primaria in ogni tabella e alcune righe di dati di esempio. Nella procedura successiva verrà illustrata la modalità di visualizzazione delle informazioni in un'applicazione. Verrà creata anche una chiave esterna per specificare quali record di una tabella possono corrispondere ai record dell'altra tabella.

CREAZIONE DELLA TABELLA STUDENTI

1. In **Esplora server** o **Esplora database** espandere il nodo **Connessioni dati**, quindi il nodo **Database1.mdf**.

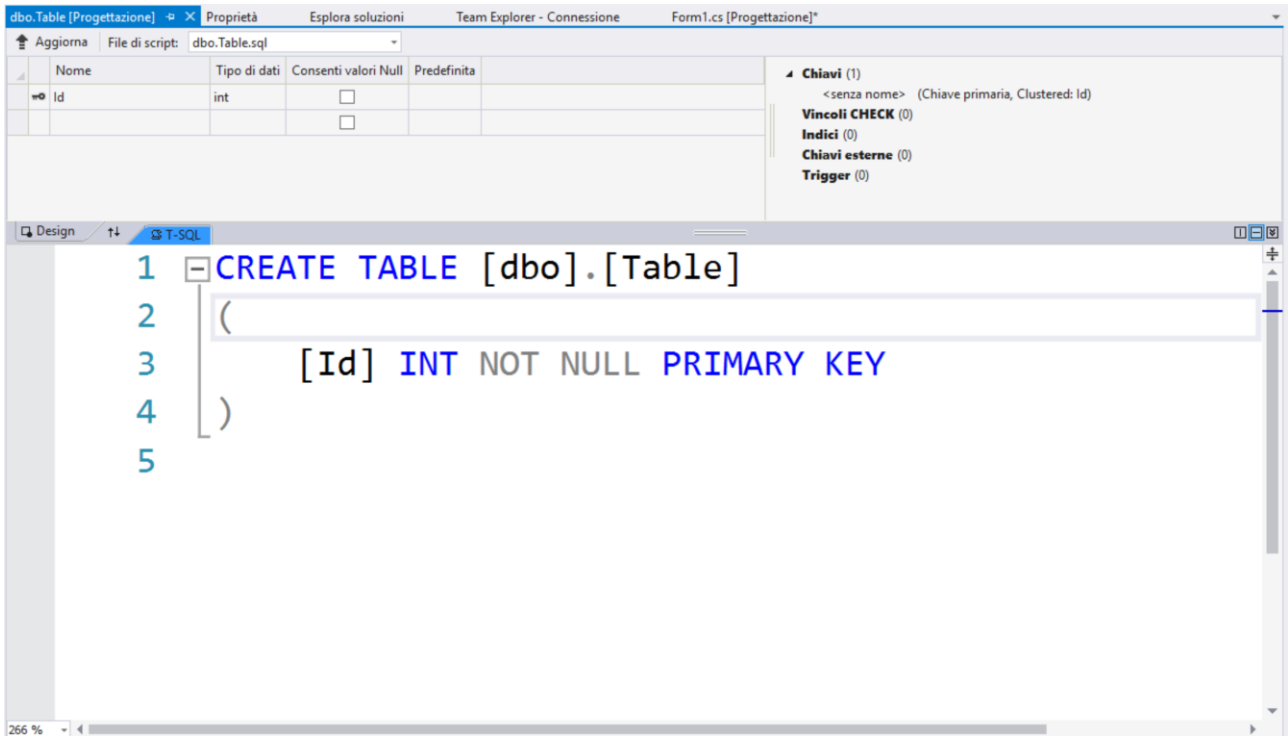


Se la finestra di esplorazione per la versione di Visual Studio non è aperta, scegliere **Visualizza, Esplora server** oppure sulla barra dei menu scegliere **Visualizza, Altre finestre, Esplora database**.

2. Aprire il menu di scelta rapida per **Tabelle**, quindi scegliere **Aggiungi nuova tabella**.



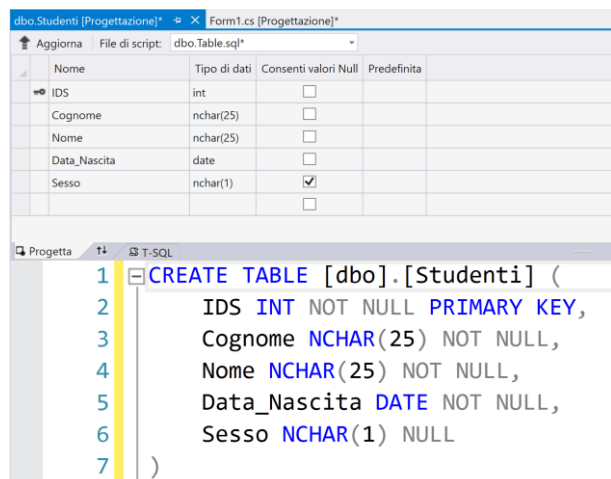
Verrà aperta **Progettazione tabelle** e verrà visualizzata una griglia con una riga predefinita, che rappresenta una singola colonna della tabella che si sta creando. Aggiungendo righe alla griglia, vengono aggiunte colonne alla tabella.



La finestra è divisa in due sezioni sincronizzate. La parte alta è visuale e permette di costruire la tabella graficamente, mentre la sezione bassa permette di dichiarare la tabella col comando SQL. Il comando SQL è CREATE TABLE, la cui sintassi è analizzata nei dettagli in un'altra dispensa del corso.

Possiamo costruire la seguente tabella a riga di comando:

```
CREATE TABLE [dbo].[Studenti]
(
    IDS INT NOT NULL PRIMARY KEY,
    Cognome NCHAR(25) NOT NULL,
    Nome NCHAR(25) NOT NULL,
    Data_Nascita DATE NOT NULL,
    Sesso NCHAR(1) NULL
)
```



Per confermare la tabella dobbiamo fare clic sul pulsante aggiorna sopra la tabella appena definita (vedi figura a lato)

4. Nella finestra di dialogo **Anteprima aggiornamenti database** scegliere il pulsante **Aggiorna database**. Le modifiche vengono salvate nel file del database locale.



CREAZIONE DELLA TABELLA OPERE

Analogamente alla precedente procedura creare una nuova tabella denominata Opere:

```
CREATE TABLE [dbo].[Opere]
(
    [IDO] INT NOT NULL PRIMARY KEY,
    [Titolo] CHAR(20) NOT NULL,
    [ISBN] NCHAR(16) NOT NULL
)
```

CREAZIONE DELLA TABELLA TESTI

Analogamente alla precedente procedura creare una nuova tabella denominata Testi:

```
CREATE TABLE [dbo].[Testi]
(
    [IDT] INT NOT NULL,
    [Posizione] NCHAR(10) NULL,
    [ID_Opera_FK] INT NULL,
    CONSTRAINT [PK_Table] PRIMARY KEY ([IDT])
)
```

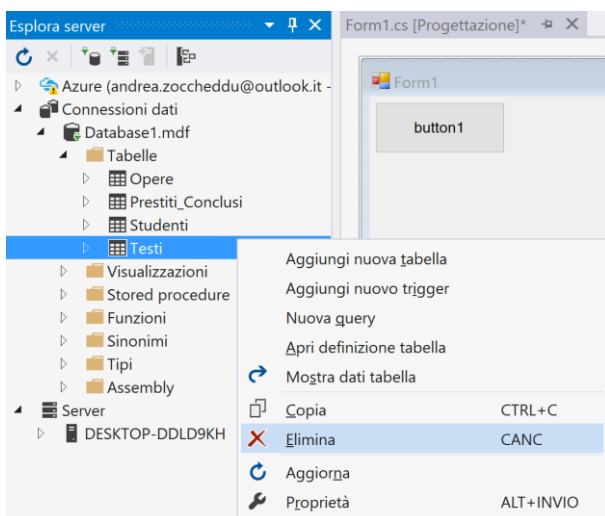
CREAZIONE DELLA TABELLA PRESTITI_CONCLUSI

Analogamente alla precedente procedura creare una nuova tabella denominata Prestiti_Conclusi:

```
CREATE TABLE [dbo].[Prestiti_Conclusi]
(
    [IDPC] INT NOT NULL PRIMARY KEY,
    [Data] DATE NOT NULL,
    [Data_Scadenza] DATE NOT NULL,
    [Data_Restituzione] DATE NOT NULL,
    [ID_Studente_FK] INT NOT NULL,
    [ID_Testo_FK] INT NOT NULL
)
```

PER ELIMINARE UNA TABELLA SBAGLIATA

Nel caso abbiate creato una tabella sbagliata è possibile eliminarla con il seguente comando:

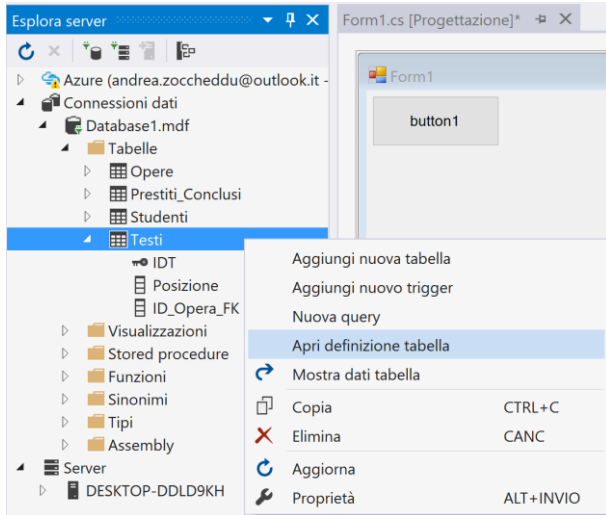


1. Selezionare la tabella da eliminare col mouse
2. Fare clic destro per visualizzare il menu contestuale
3. Scegliere Elimina e confermare



PER CREARE UNA CHIAVE ESTERNA

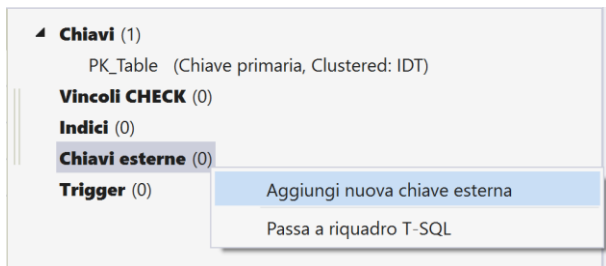
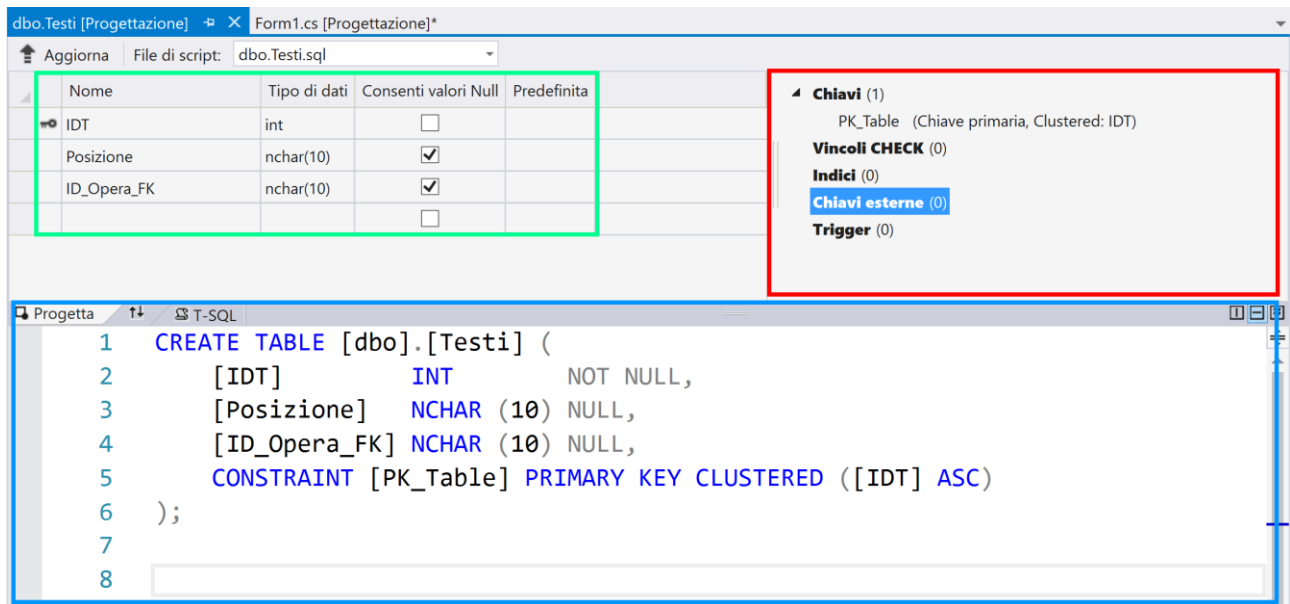
Abbiamo creato delle tabelle corredate di chiave primaria. Adesso proviamo a collegare i campi delle chiavi esterne ai campi chiave primaria per instaurare le chiavi esterne.



Intanto, aggiorniamo il database per visualizzare tutte le tabelle esistenti. Poi, scegliamo una tabella su cui impostare una chiave esterna; nel nostro esempio inizieremo dalla tabella Testi che deve essere collegata alla tabella Opere.

Selezioniamo, quindi, la tabella Testi e visualizziamo il menu contestuale col clic destro del mouse; poi scegliamo la voce Apri definizione tabella per visualizzare la struttura.

Si può osservare che la finestra di progettazione è sempre composta di due sezioni e che, nella sezione alta, vi è un riquadro sulla destra (nell'immagine è bordata di rosso):



Sulla voce Chiavi esterne possiamo fare clic destro per visualizzare il menu contestuale

Dal menu contestuale possiamo scegliere la voce Aggiungi Nuova Chiave Esterna



▲ **Chiavi** (1)
 PK_Table (Chiave primaria, Clustered: IDT)
Vincoli CHECK (0)
Indici (0)
 ▲ **Chiavi esterne** (1)
 FK_Testi_Opere
Trigger (0)

Il comando crea una casella di testo con una stringa che è il nome del vincolo di chiave esterna

Possiamo modificare il nome di chiave esterna per adeguarla alla nostra situazione; la denominiamo: **FK_Testi_Opere**

Confermiamo con Invio

Possiamo osservare che il comando ha creato una nuova riga nello script del comando CREATE TABLE che corrisponde al vincolo di chiave esterna

```
CREATE TABLE [dbo].[Testi] (
  [IDT]          INT          NOT NULL,
  [Posizione]   NCHAR (10) NULL,
  [ID_Opera_FK] NCHAR (10) NULL,
  CONSTRAINT [PK_Table] PRIMARY KEY CLUSTERED ([IDT] ASC),
  CONSTRAINT [FK_Testi_Opere]
    FOREIGN KEY ([Column])
    REFERENCES [ToTable]([ToTableColumn])
);
```

La riga è scorretta perché manca ancora la specifica di quale campo vincolare e quale tabella esterna correlare e infine quale campo della tabella esterna correlare.

```
CONSTRAINT [FK_Testi_Opere] FOREIGN KEY ([Column]) REFERENCES [ToTable]([ToTableColumn])
```

La clausola va modificata come segue:

```
CONSTRAINT [FK_Testi_Opere] FOREIGN KEY ([ID_Opera_FK]) REFERENCES [Opere]([IDO])
```

Il comando CREATE TABLE, alla fine, dovrebbe assomigliare al seguente:

```
CREATE TABLE [dbo].[Testi] (
  [IDT]          INT          NOT NULL,
  [Posizione]   NCHAR (10) NULL,
  [ID_Opera_FK] INT          NULL,
  CONSTRAINT [PK_Table]
    PRIMARY KEY CLUSTERED ([IDT] ASC),
  CONSTRAINT [FK_Testi_Opere]
    FOREIGN KEY ([ID_Opera_FK])
    REFERENCES [Opere]([IDO])
);
```



Nome	Tipo di dati	Consenti valori Null	Predefinita
IDT	int	<input type="checkbox"/>	
Posizione	nchar(10)	<input checked="" type="checkbox"/>	
ID_Opera_FK	int	<input checked="" type="checkbox"/>	

```

1 CREATE TABLE [dbo].[Testi] (
2     [IDT]          INT          NOT NULL,
3     [Posizione]   NCHAR (10)  NULL,
4     [ID_Opera_FK] INT          NULL,
5     CONSTRAINT [PK_Table] PRIMARY KEY CLUSTERED ([IDT] ASC),
6     CONSTRAINT [FK_Testi_Opere]
7         FOREIGN KEY ([ID_Opera_FK])
8             REFERENCES [Opere]([ID0])
9 );
10
11

```

Confermare la tabella con il pulsante **Aggiorna**, come al solito. Nella finestra di dialogo **Anteprima aggiornamenti database** scegliere il pulsante **Aggiorna database**. Le modifiche saranno salvate nel file del database locale.

La tabella Testi deve essere corredata con una seconda chiave esterna che lo collega agli Studenti, nel caso un testo sia dato in prestito a quest'ultimo.

Se non è ancora aperta, apriamo la tabella testi in modalità modifica ed aggiungiamo un campo:

```

CREATE TABLE [dbo].[Testi] (
    [IDT]          INT          NOT NULL,
    [Posizione]   NCHAR (10)  NULL,
    [ID_Opera_FK] INT          NULL,
    [ID_Prestato_a_FK] INT NULL,
    CONSTRAINT [PK_Table] PRIMARY KEY CLUSTERED ([IDT] ASC),
    CONSTRAINT [FK_Testi_Opere] FOREIGN KEY ([ID_Opera_FK]) REFERENCES [dbo].[Opere]
    ([ID0])
);

```

poi imponiamo una chiave esterna sul campo:

```

CONSTRAINT [FK_Testi_Studenti]
    FOREIGN KEY ([ID_Prestato_a_FK])
    REFERENCES [Studenti](IDS)

```

La tabella Testi diventa come segue:



Nome	Tipo di dati	Consenti valori Null	Predefinita
IDT	int	<input type="checkbox"/>	
Posizione	nchar(10)	<input checked="" type="checkbox"/>	
ID_Opera_FK	int	<input checked="" type="checkbox"/>	
ID_Prestato_a_FK	int	<input checked="" type="checkbox"/>	

```

1 CREATE TABLE [dbo].[Testi] (
2     [IDT] INT NOT NULL,
3     [Posizione] NCHAR (10) NULL,
4     [ID_Opera_FK] INT NULL,
5     [ID_Prestato_a_FK] INT NULL,
6     CONSTRAINT [PK_Table] PRIMARY KEY CLUSTERED ([IDT] ASC),
7     CONSTRAINT [FK_Testi_Opere]
8     FOREIGN KEY ([ID_Opera_FK])
9     REFERENCES [dbo].[Opere] ([IDO]),
10    CONSTRAINT [FK_Testi_Studenti]
11    FOREIGN KEY ([ID_Prestato_a_FK])
12    REFERENCES [Studenti](IDS)
13 );
14
15
    
```

Premiamo aggiorna per lanciare il comando; poi aggiorniamo il database.

Chiudiamo la finestra di Testi

Adesso impostiamo le chiavi esterne per la tabella Prestiti_Conclusi.

Selezioniamo la tabella Prestiti_Conclusi e apriamola.

Impostiamo la prima chiave esterna verso Studenti

Impostiamo la seconda chiave esterna verso Testi





Ecco la tabella alla fine delle modifiche:

The screenshot shows the SQL Server Enterprise Designer interface. The top pane displays the table design for 'Prestiti_Conclusi' with the following columns:

Nome	Tipo di dati	Consenti valori Null	Predefinita
IDPC	int	<input type="checkbox"/>	
Data	date	<input type="checkbox"/>	
Data_Scadenza	date	<input type="checkbox"/>	
Data_Restituzione	date	<input type="checkbox"/>	
ID_Studente_FK	int	<input type="checkbox"/>	
ID_Testo_FK	int	<input type="checkbox"/>	

The right pane shows the table's properties:

- Chiavi (1)**: <senza nome> (Chiave primaria, Clustered: IDPC)
- Vincoli CHECK (0)**
- Indici (0)**
- Chiavi esterne (2)**:
 - FK_Prestiti_Conclusi_Studente (IDS)
 - FK_Prestiti_Conclusi_Testo (IDT)
- Trigger (0)**

The bottom pane shows the T-SQL script for creating the table:

```

1 CREATE TABLE [dbo].[Prestiti_Conclusi] (
2     [IDPC] INT NOT NULL,
3     [Data] DATE NOT NULL,
4     [Data_Scadenza] DATE NOT NULL,
5     [Data_Restituzione] DATE NOT NULL,
6     [ID_Studente_FK] INT NOT NULL,
7     [ID_Testo_FK] INT NOT NULL,
8     PRIMARY KEY CLUSTERED ([IDPC] ASC),
9     CONSTRAINT [FK_Prestiti_Conclusi_Studente]
10    FOREIGN KEY ([ID_Studente_FK])
11    REFERENCES Studenti([IDS]),
12    CONSTRAINT [FK_Prestiti_Conclusi_Testo]
13    FOREIGN KEY ([ID_Testo_FK])
14    REFERENCES Testi([IDT])
15 );
16

```

The screenshot shows the 'Esplora server' window in SQL Server Enterprise Explorer. The tree view is expanded to show the 'Biblioteca.mdf' database. The 'Tabelle' folder is expanded, showing the following tables:

- Opere
 - IDO
 - Titolo
 - ISBN
- Prestiti_Conclusi
 - IDPC
 - Data
 - Data_Scadenza
 - Data_Restituzione
 - ID_Studente_FK
 - ID_Testo_FK
- Studenti
 - IDS
 - Cognome
 - Nome
 - Data_Nascita
 - Sesso
- Testi
 - IDT
 - Posizione
 - ID_Opera_FK
 - ID_Prestato_A

Alla fine, fare clic su Aggiorna e poi aggiornare anche il database.

Nella finestra Esplora Server l'elenco delle tabelle dovrebbe risultare simile alla figura qui proposta.

Assicurarsi che le chiavi esterne siano ben correlate prima di inserire eventuali dati nelle tabelle.



INDICI SUI CAMPI DELLE TABELLE

PER VINCOLARE E ANCHE IMPOSTARE DUPLICATI NON AMMESSI

INDICI, ESEMPIO 1

The screenshot shows the 'Students' table in SQL Server Enterprise Manager. The 'Matricola' field is highlighted in blue. The 'Keys' pane on the right shows the primary key 'IDS'. The 'Indexes' pane is empty. The 'SQL' pane shows the table definition and the creation of a clustered index on the 'IDS' field.

Nome	Tipi di dati	Consenti valori Null	Predefinita
IDS	int	<input type="checkbox"/>	
Cognome	nchar(25)	<input type="checkbox"/>	
Nome	nchar(25)	<input type="checkbox"/>	
Data_Nascita	date	<input type="checkbox"/>	
Sesso	nchar(1)	<input checked="" type="checkbox"/>	
Matricola	char(10)	<input checked="" type="checkbox"/>	

```

1 CREATE TABLE [dbo].[Studenti] (
2     [IDS] INT NOT NULL,
3     [Cognome] NCHAR (25) NOT NULL,
4     [Nome] NCHAR (25) NOT NULL,
5     [Data_Nascita] DATE NOT NULL,
6     [Sesso] NCHAR (1) NOT NULL,
7     [Matricola] CHAR(10) NULL,
8     PRIMARY KEY CLUSTERED ([IDS] ASC)
9 );
10
11
12 GO
13
14 CREATE INDEX [VincoloMatricola] ON [dbo].[Studenti] ([Matricola])
15

```

1. In **Esplora server** o **Esplora database**, selezionare la tabella Studenti
2. Aprire la definizione della tabella

3. Inserire un nuovo campo Matricola
4. Selezionare la voce Indici sulla destra

5. Viene aggiunto un comando per la creazione di un indice
6. Assegnare un nome all'indice
7. Modificare il campo cui fa riferimento il vincolo

8. Usare Matricola come campo indicizzato

`CREATE INDEX [VincoloMatricola] ON [dbo].[Studenti] ([Matricola])`

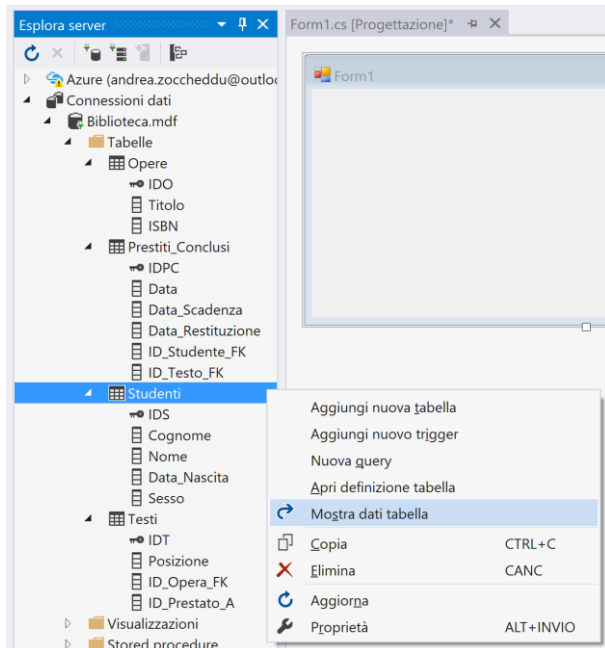
Se si vuole creare un indice che non ammette duplicati usare invece il comando:

```
CREATE UNIQUE NONCLUSTERED INDEX [VincoloMatricola]
ON [dbo].[studenti] ([Matricola] ASC )
```



INSERIMENTO DI DATI NELLE TABELLE

PER INSERIRE DATI NELLE TABELLE



9. In **Esplora server** o **Esplora database**, espandere il nodo per il database di esempio.
10. Aprire il menu di scelta rapida per il nodo **Tabelle**, scegliere **Aggiorna**, quindi espandere il nodo **Tabelle**.
11. Aprire il menu di scelta rapida per la tabella **Studenti** e scegliere **Mostra dati tabella**.
12. Aggiungere i dati desiderati per almeno tre studenti.

IDS	Cognome	Nome	Data_Nascita	Sesso
1	CINESCA	SARA	01/04/2000	F
2	CORRENTE ...	DARIA	25/04/2000	F
3	PIANO	GUIDO	25/12/2000	M
4	PIANO	REMO	31/12/2000	M
* NULL	NULL	NULL	NULL	NULL

Dopo aver digitato i dati nelle righe della tabella è possibile visualizzare il comando SQL che genera questo inserimento; premere l'icona dello script, in alto a destra sopra la tabella (bordata di rosso nella figura). I comandi generati sono i seguenti:

```
INSERT INTO [dbo].[Studenti] ([IDS], [Cognome], [Nome], [Data_Nascita], [Sesso])
VALUES (1, N'CINESCA', N'SARA', N'2000-04-01', N'F')
INSERT INTO [dbo].[Studenti] ([IDS], [Cognome], [Nome], [Data_Nascita], [Sesso])
VALUES (2, N'CORRENTE', N'DARIA', N'2000-04-25', N'F')
INSERT INTO [dbo].[Studenti] ([IDS], [Cognome], [Nome], [Data_Nascita], [Sesso])
VALUES (3, N'PIANO', N'GUIDO', N'2000-12-25', N'M')
INSERT INTO [dbo].[Studenti] ([IDS], [Cognome], [Nome], [Data_Nascita], [Sesso])
VALUES (4, N'PIANO', N'REMO', N'2000-12-31', N'M')
```

La sintassi del comando è analizzata in un'altra dispensa.

Chiudere la tabella dei dati per gli **Studenti** e prepararsi a inserire delle **Opere**. I valori **11** e **12** sono le due chiavi primarie per le opere inserite nell'esempio.

Chiudere la tabella **Opere** e prepararsi ad inserire dati nei **Testi**.

Il valore **11** inserito nella colonna **ID_Opera_FK** indica che queste sono le copie dell'opera «**La Divina Commedia**» della biblioteca.

Analogamente il **12** indica le copie dell'opera «**I Promessi Sposi**».

Le colonne **Data_Prestito** e **Data_Scadenza** sono stati aggiunti dopo.

IDO	Titolo	ISBN
11	La Divina Commedia	121-545-656
12	I Promessi Sposi	414-252-366
* NULL	NULL	NULL

IDT	Posizione	ID_Opera_FK	ID_Prestato_A	Data_Prestito	Data_Scadenza
21	P001-A12	11	1	12/01/2011	12/06/2011
22	P001-A13	11	2	14/02/2011	14/06/2011
23	P001-A14	11	NULL	NULL	NULL
24	P321-B72	12	NULL	NULL	NULL
25	P321-B75	12	NULL	NULL	NULL
26	P321-B77	12	1	12/01/2011	12/06/2011
* NULL	NULL	NULL	NULL	NULL	NULL



Chiudere la tabella Testi e prepararsi ad inserire dati nei Prestiti_Conclusi.

ID	Data_Inizio	Data_Restituzione	ID_Studente_FK	ID_Testo_FK	
31	04/03/2010	04/04/2010	14/03/2010	1	21
32	14/03/2010	14/04/2010	25/04/2010	1	24
33	15/03/2010	15/04/2010	25/04/2010	2	22
34	17/03/2010	17/04/2010	22/03/2010	2	25
NU...	NULL	NULL	NULL	NULL	NULL

Importante: Verificare che tutti i valori delle chiavi primarie e delle chiavi esterne siano interi e che ogni chiave esterna corrisponda a una chiave primaria già esistente.

- Sulla barra dei menu scegliere **File, Salva tutto.**
- Sulla barra dei menu scegliere **File, Chiudi soluzione.**

CONNESSIONE AI DATI IN UN FILE DI DATABASE LOCALE

È possibile visualizzare i dati in modo semplice e rapido da un file di database locale nell'applicazione creando un dataset e aggiungendo controlli associati a dati all'applicazione. In questa procedura verranno visualizzati i dati del file di database locale creato.

Dopo aver creato un progetto Windows Form, verrà stabilita la connessione al database e verrà indicato di visualizzare i dati della tabella Clienti in una griglia dati del form dell'applicazione.

In questa procedura sono incluse le attività seguenti:

- Creazione e configurazione di un dataset
- Aggiunta di controlli con associazione a dati

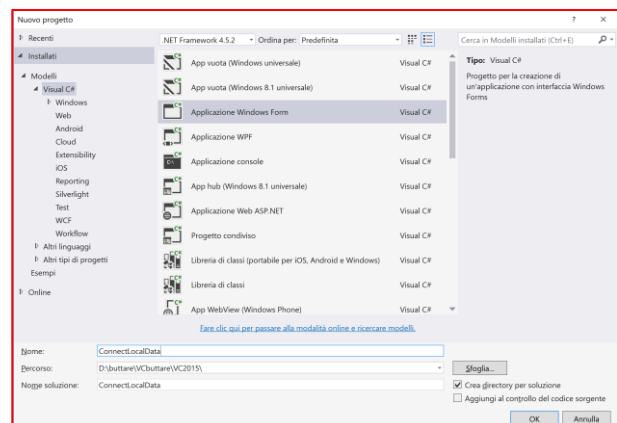
PREREQUISITI

Per completare questa procedura dettagliata, è necessario accedere al database **Database1.mdf** creato completando la procedura sopra descritta.

CREAZIONE E CONFIGURAZIONE DI UN DATASET

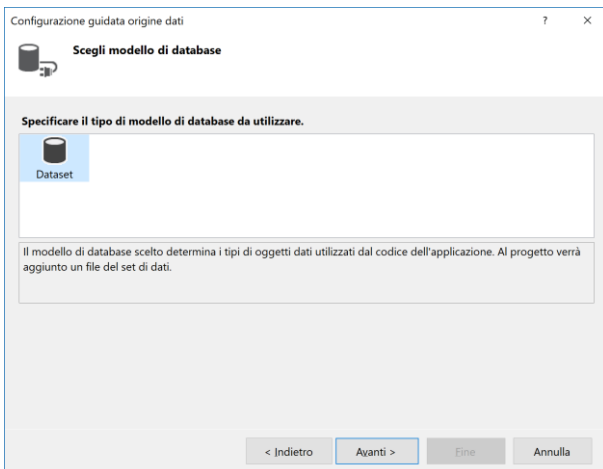
PER CREARE E CONFIGURARE UN DATASET

1. Creare un progetto Windows Form e denominarlo **ConnectLocalData**.

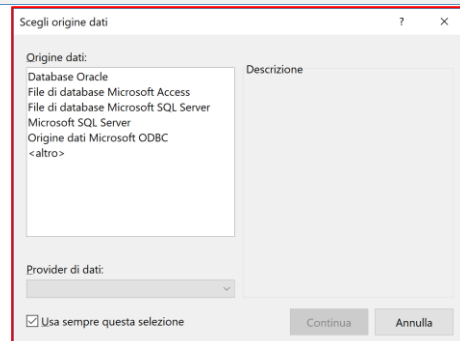
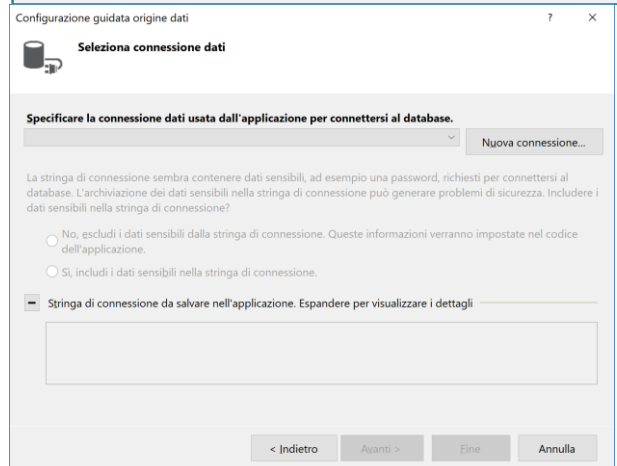
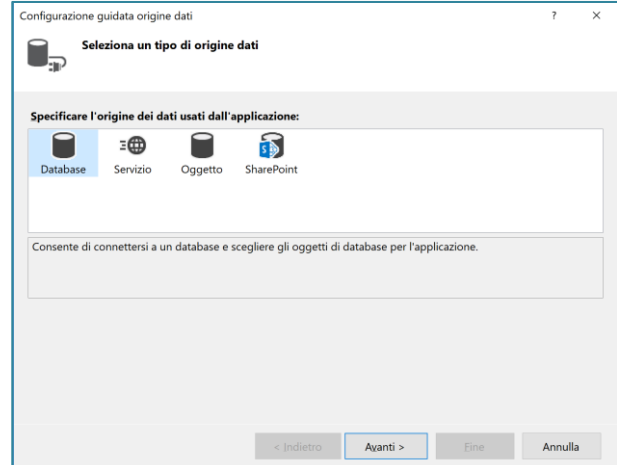
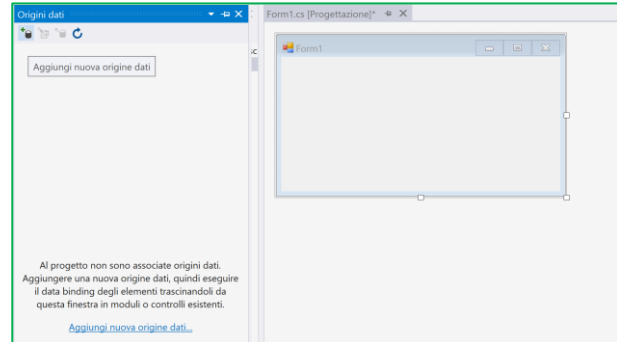




2. Se la finestra **Origini dati** non è visualizzata, scegliere la combinazione di tasti MAIUSC-ALT-D oppure sulla barra dei menu scegliere **Visualizza, Altre finestre e Mostra origini dati**.
3. Nella finestra **Origini dati** scegliere il collegamento **Aggiungi nuova origine dati**.
4. In **Configurazione guidata origine dati** scegliere **Avanti** due volte per accettare le impostazioni predefinite.



5. Nella pagina **Selezione connessione dati** scegliere **Nuova connessione**.
6. Verrà visualizzata la finestra di dialogo **Scegli origine dati**.



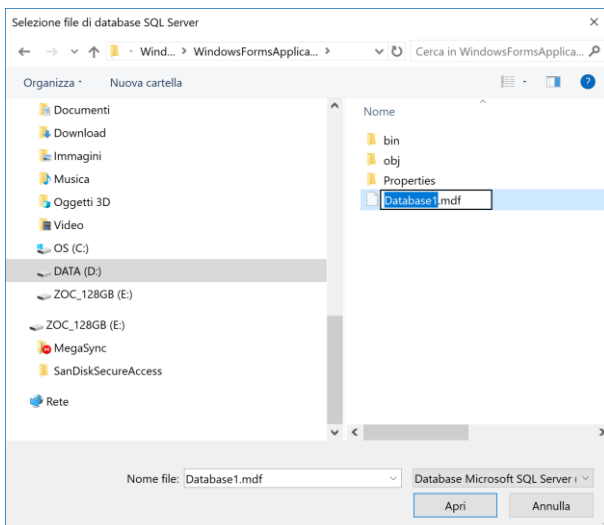


7. Nell'elenco **Origine dati** scegliere **File di database Microsoft SQL Server**, quindi scegliere **Continua**.

8. Verrà visualizzata la finestra di dialogo **Aggiungi connessione**.

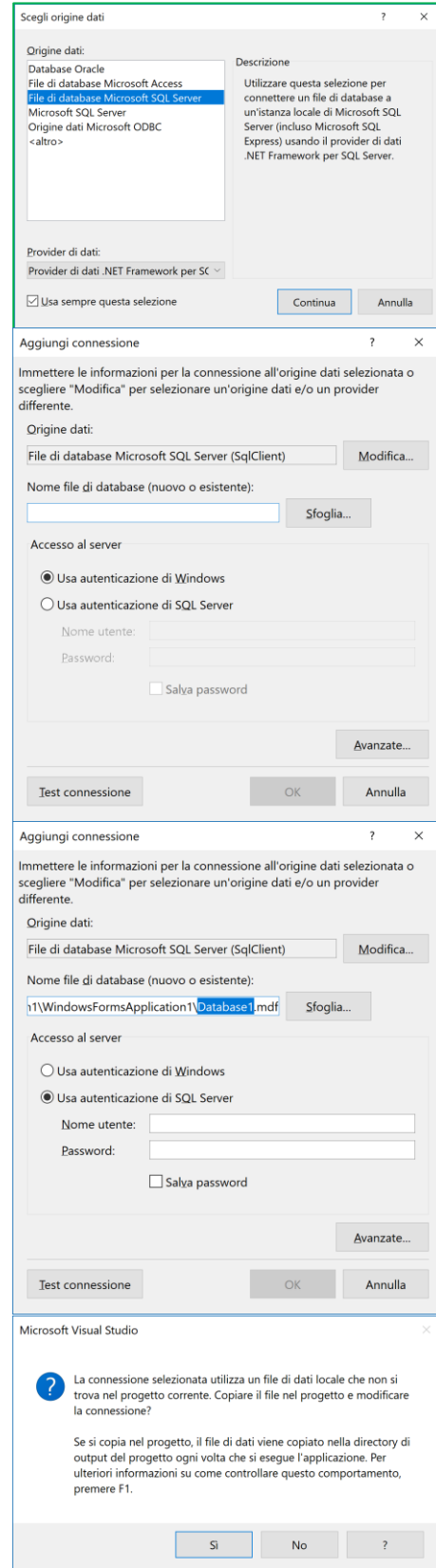
9. Nella casella **Nome file di database** specificare il file creato precedentemente

10. In **Accesso al server** accettare i valori predefiniti, scegliere **OK**, quindi scegliere **Avanti**.



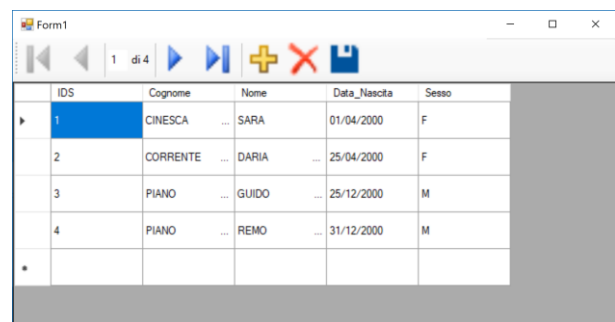
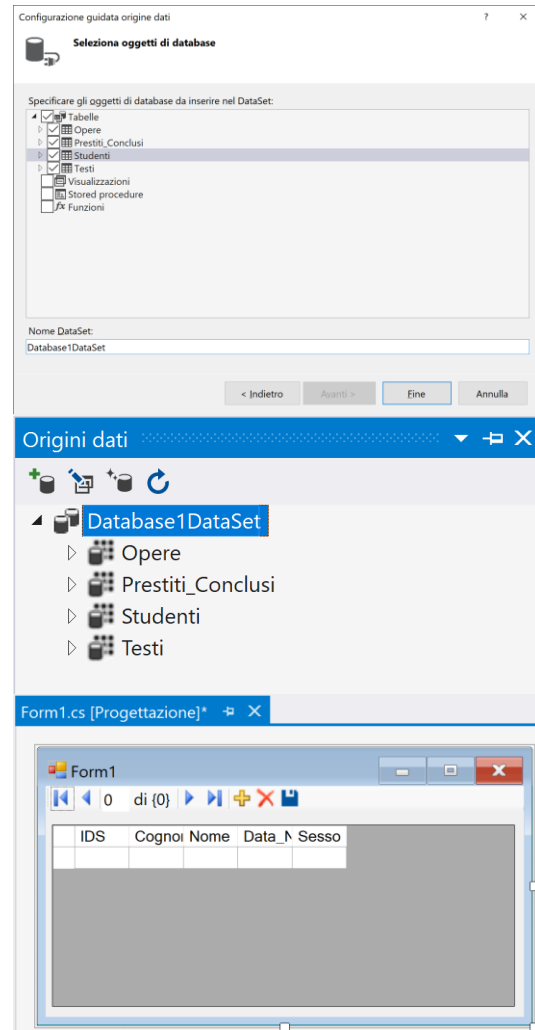
Nota: Quando si creano connessioni a un file di database locale, è possibile scegliere di creare una copia del database nel progetto o di eseguire la connessione al file di database esistente nella posizione attuale.

11. Nella finestra di dialogo visualizzata scegliere **Sì** per copiare il file di database nel progetto.





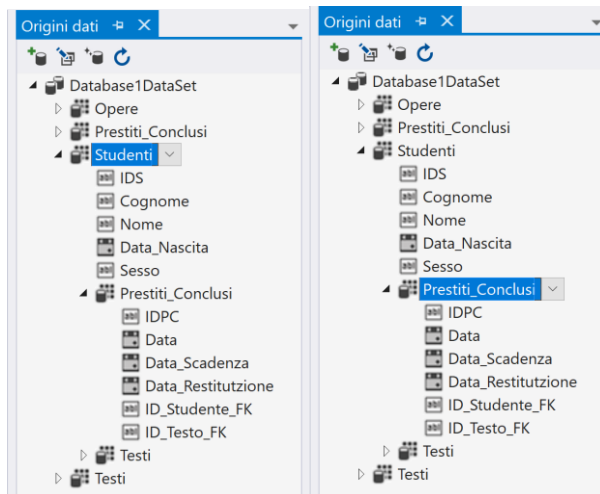
- 12.** Nella pagina **Salva stringa di connessione nel file di configurazione dell'applicazione** scegliere **Avanti**.
- 13.** Nella pagina **Selezione oggetti di Database** espandere il **Tabelle** nodo, seleziona il **clienti** e **ordini** caselle di controllo e quindi scegliere il **Fine** pulsante.
- 14.** Adesso la finestra Origini dati mostra l'oggetto Database1DataSet che al suo interno mostra le tabelle con cui può comunicare
- 15.** Selezionare la tabella Studenti e trascinarla (Drag_and_Drop) nel Form1
- Sul form vengono visualizzati un oggetto [DataGridView](#) e un controllo ToolStrip ([BindingNavigator](#)) per lo spostamento all'interno dei record.
Nella barra dei componenti vengono visualizzati gli oggetti [SampleDatabaseDataSet](#), [CustomersTableAdapter](#), [BindingSource](#) e [BindingNavigator](#).
- 16.** Salvare tutto ed avviare l'applicazione





VISUALIZZARE DATI CORRELATI IN UN'APPLICAZIONE WINDOWS FORM

Per visualizzare i dati correlati è possibile trascinare gli elementi che condividono lo stesso nodo principale della tabella dalla **Origini dati** (finestra) nel form in uso. Se ad esempio si dispone di un'origine dati in cui sono presenti una tabella Studenti e una tabella correlata Prestiti, è possibile visualizzare entrambe le tabelle come nodi di livello principale nella visualizzazione ad albero all'interno della finestra **Origini dati**. Espandendo il nodo Studenti vengono visualizzate le colonne e si nota che l'ultima colonna dell'elenco è un nodo espandibile che rappresenta la tabella **Prestiti_Conclusi**.



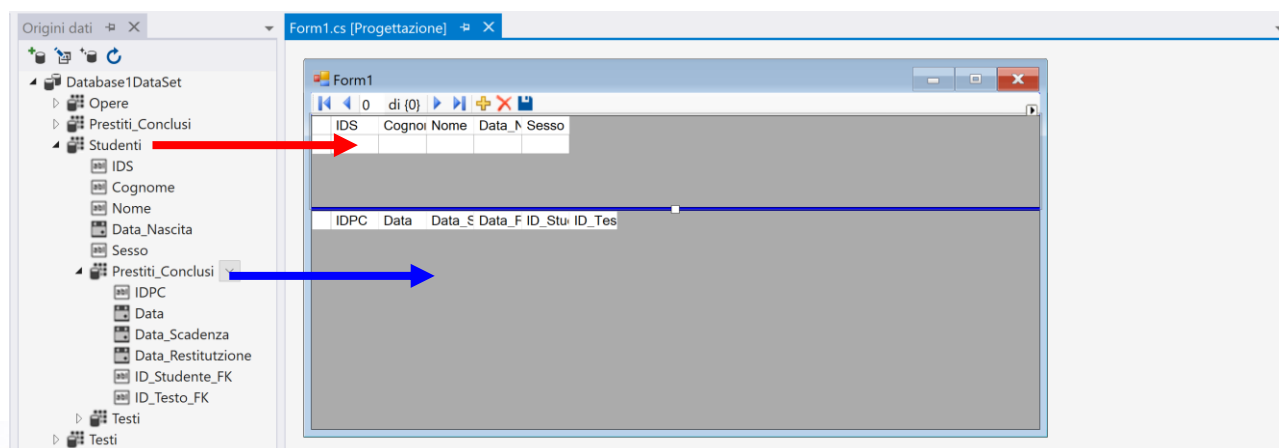
Il nodo indica i prestiti relativi a uno studente. Questo consente di creare un form1 che consenta di selezionare uno studente della griglia e in conseguenza visualizzare in un'altra griglia l'elenco dei prestiti conclusi relativi a tale soggetto.

Per farlo sarà opportuno trascinare prima un elemento e poi l'altro (il nodo correlato) elemento da questa singola gerarchia nel form1.

CREAZIONE DI CONTROLLI ASSOCIATI A DATI PER LA VISUALIZZAZIONE DI RECORD CORRELATI

PER CREARE CONTROLLI PER LA VISUALIZZAZIONE DI RECORD CORRELATI

1. Aprire il form in Windows Forms Designer.
2. Aprire la finestra **Origini dati**.
3. Espandere il nodo che rappresenta la tabella padre nella relazione (la tabella padre è quella sul lato "uno" di una relazione uno-a-molti).
4. Trascinare prima il nodo Studenti nella parte superiore del Form1 (linea rossa)
5. Trascinare poi il sotto-nodo Prestiti_Studenti nella parte inferiore del Form1 (linea blu)





6. Le tabelle figlio correlate vengono visualizzate come nodi espandibili alla fine dell'elenco di colonne della tabella padre. Trascinare gli elementi da visualizzare da un nodo correlato nel form in uso.
7. È possibile usare le proprietà Dock delle griglie per occupare parte superiore ed inferiore del Form1
8. Avviare il programma e provare a selezionare i vari studenti

IDS	Cognome	Nome	Data_Nascita	Sesso	IDPC	Data	Data_Scadenza	Data_Restituzione	ID_Studente_FK	ID_T
1	CINESCA	SARA	01/04/2000	F						
31			04/03/2010		04/04/2010	14/03/2010	1		21	
32			14/03/2010		14/04/2010	20/04/2010	1		22	

DML DI SQL SERVER

QUERY

PREPARAZIONE DI UNA QUERY SU UNA TABELLA

Selezioniamo la finestra Esplora Server per visualizzare le tabelle presenti nel database

Selezioniamo la tabella Studenti e facciamo un clic destro su essa per visualizzare il menu contestuale relativo alla tabella

Nel menu possiamo fare clic sulla voce **Nuova query**

Compare una finestra di dialogo per costruire una query SQL in modalità testuale

È possibile quindi scrivere la query desiderata, osservando che sulla paletta è comparsa una barra di pulsanti con varie funzioni

Esplora server

dbo.Studenti [Dati] Form1.cs [Progettazione]

IDS	Cognome	Nome	Data_Nascita	Sesso
1	CINESCA	SARA	01/04/2000	F
2	CORRENTE	DARIA	25/04/2000	F
3	PIANO	GUIDO	25/12/2000	M
4	PIANO	REMO	31/12/2000	M
* NULL	NULL	NULL	NULL	NULL

SQLQuery1.sql

```

1 SELECT *
2 FROM Studenti;
```

D:\BUTTARE\VCBUTTARE\VC2015\

Il primo dei pulsanti consente l'esecuzione della query.



The first screenshot shows a query window with the following SQL code:

```
1 SELECT *
2 FROM Studenti;
```

The results pane shows the following data:

IDS	Cognome	Nome	Data_Nascita	Sesso
1	CINESCA	SARA	2000-04-01	F
2	CORRENTE	DARIA	2000-04-25	F
3	PIANO	GUIDO	2000-12-25	M
4	PIANO	REMO	2000-12-31	M

The second screenshot shows the same query window with the following SQL code:

```
1 SELECT *
2 FROM Studenti
3 WHERE Sesso = 'F';
```

The results pane shows the following data:

IDS	Cognome	Nome	Data_Nascita	Sesso
1	CINESCA	SARA	2000-04-01	F
2	CORRENTE	DARIA	2000-04-25	F

PREPARAZIONE DI UNA QUERY SU PIÙ TABELLE

Selezioniamo la finestra Esplora Server per visualizzare le tabelle presenti nel database

Selezioniamo la voce tabelle e col clic destro possiamo visualizzare il menu contestuale su cui possiamo fare clic sulla voce **Nuova query**

The screenshot shows the 'Esplora server' window with the 'Tabelle' folder selected. The context menu is open, showing options like 'Nuova query', 'Aggiorna', and 'Proprietà'. The 'Nuova query' option is highlighted.

Compare una finestra di dialogo per costruire una query SQL in modalità testuale

È possibile quindi scrivere la query desiderata, che possiamo eseguire col pulsante verde:

The screenshot shows a query window with the following SQL code:

```
1 SELECT *
2 FROM Studenti INNER JOIN Prestiti_Conclusi
3 ON Studenti.IDS = Prestiti_Conclusi.ID_Studente_FK
4 ORDER BY Prestiti_Conclusi.Data;
```

The results pane shows the following data:

IDS	Cognome	Nome	Data_Nascita	Sesso	IDPC	Data	Data_Scadenza	Data_Restituzione	ID_Studente_FK	ID_Testo_FK
1	CINESCA	SARA	2000-04-01	F	32	2010-03-04	2010-04-04	2010-03-14	1	21
2	CINESCA	SARA	2000-04-01	F	32	2010-03-14	2010-04-14	2010-04-20	1	22
3	CORRENTE	DARIA	2000-04-25	F	33	2010-03-15	2010-04-15	2010-03-22	2	21
4	CORRENTE	DARIA	2000-04-25	F	34	2010-04-25	2010-05-25	2010-05-13	2	23



VISTE

La differenza tra query e vista è analizzata in un'altra dispensa. in questa sede ci limiteremo ad osservare che la vista è una query che può essere memorizzata con un nome. La vista resta quindi nella base di dati e potrà essere riutilizzata al bisogno.

VISTA, ESEMPIO 1

Per creare una vista fare clic destro su Visualizzazioni

Viene mostrata una pagina per scrivere il codice SQL della vista che deve essere scritta dal programmatore

```
CREATE VIEW [dbo].[Studentesse] AS
SELECT *
FROM Studenti
WHERE Sesso = 'F';
```

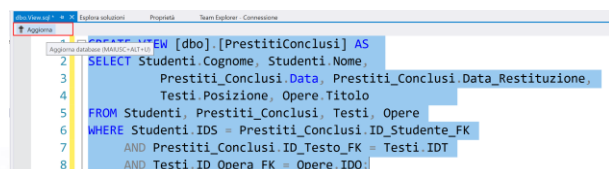
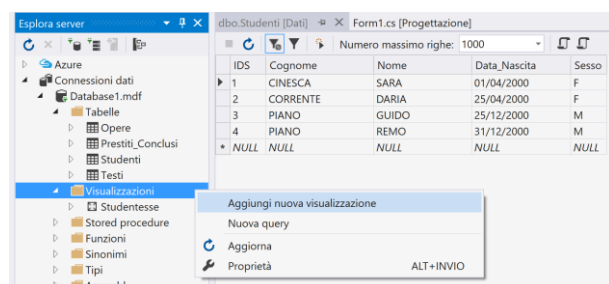
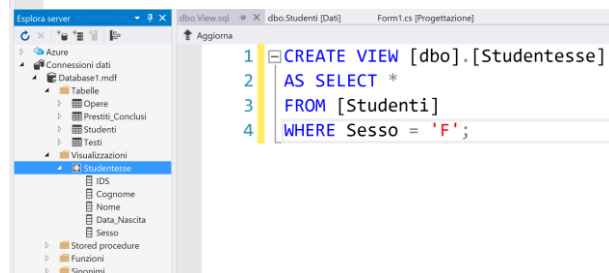
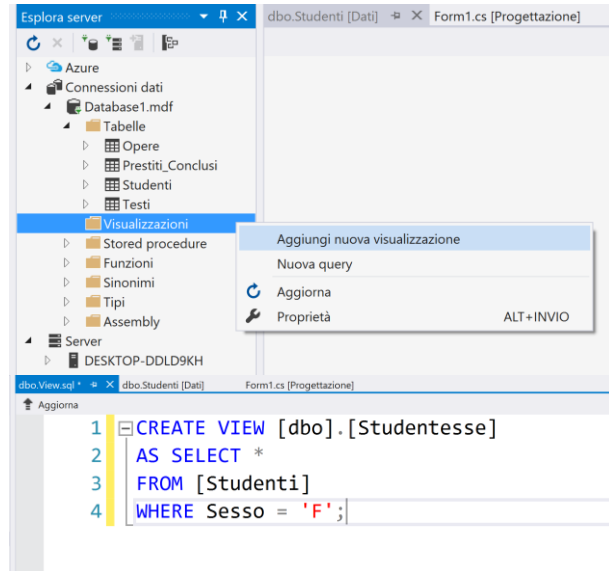
Il pulsante Aggiorna salva la vista nel database col nome scelto (Studentesse)

VISTA, ESEMPIO 2

Creare una nuova vista fare clic destro su Visualizzazioni

```
CREATE VIEW [dbo].[PrestitiConclusi] AS
SELECT Studenti.Cognome, Studenti.Nome,
        Prestiti_Conclusi.Data, Prestiti_Conclusi.Data_Restituzione,
        Testi.Posizione, Opere.Titolo
FROM Studenti, Prestiti_Conclusi, Testi, Opere
WHERE Studenti.IDS = Prestiti_Conclusi.ID_Studente_FK
AND Prestiti_Conclusi.ID_Testo_FK = Testi.IDT
AND Testi.ID_Opera_FK = Opere.IDO;
```

Confermare la Vista con Aggiorna





Aggiornare anche il database

Visualizzare e selezionare la vista desiderata

Col clic destro sulla vista visualizzare il menu contestuale

Scegliere la voce **Mostra risultati**

Viene visualizzata la tabella dinamica come risultato della query interna alla vista (adattare le colonne della query per visualizzare i dati contenuti)

In questo modo è possibile avere una tabella (dinamica) che mostra i dati utili per una visione completa della situazione.

FUNZIONI

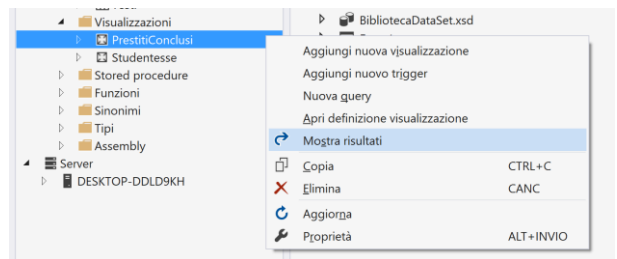
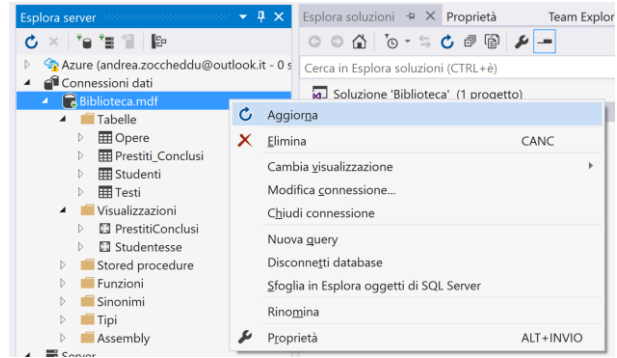
La definizione di funzione è analizzata in un'altra dispensa. Qui ci limitiamo a osservare che una funzione è un oggetto che può essere eseguito e che può rendere un valore o una tabella.

FUNZIONE, ESEMPIO 1

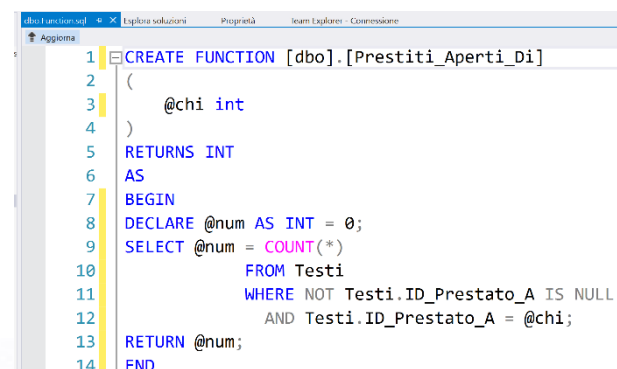
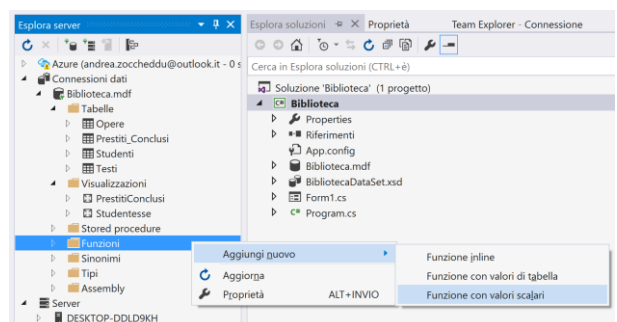
Per creare una funzione scalare fare clic destro su Funzioni, poi scegliere Aggiungi nuovo, poi scegli Funzione con valori scalari

Viene mostrata una pagina per scrivere il codice SQL della vista che deve essere scritta dal programmatore

```
CREATE FUNCTION [dbo].[Prestiti_Aperti_Di]
(
    @chi int
)
RETURNS INT
AS BEGIN
    DECLARE @num AS INT = 0;
    SELECT @num = COUNT(*)
    FROM Testi
    WHERE NOT Testi.ID_Prestato_A IS NULL
    AND Testi.ID_Prestato_A = @chi;
    RETURN @num;
END
```



Cognome	Nome	Data	Data_Restituzione	Posizione	Titolo
CINESCA	SARA	04/03/2010	14/03/2010	P001-A12	La Divina Commedia
CINESCA	SARA	14/03/2010	25/04/2010	P321-B72	I Promessi Sposi
CORRENTE	DARIA	15/03/2010	25/04/2010	P001-A13	La Divina Commedia
CORRENTE	DARIA	17/03/2010	22/03/2010	P321-B75	I Promessi Sposi
NULL	NULL	NULL	NULL	NULL	NULL

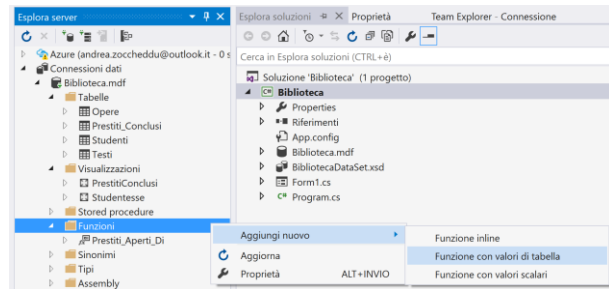




FUNZIONE, ESEMPIO 2

Per creare una funzione tabellare fare clic destro su Funzioni, poi scegliere Aggiungi nuovo, poi scegli Funzione con valori di tabella

Viene mostrata la pagina per il codice SQL



```
CREATE FUNCTION [dbo].[Prestiti_chiusi_dal_al]
(
    @dal date,
    @al date
)
RETURNS TABLE
AS RETURN (
    SELECT Studenti.Cognome, Studenti.Nome,
           Prestiti_Conclusi.Data, Prestiti_Conclusi.Data_Restituzione,
           Testi.Posizione, Opere.Titolo
    FROM Studenti, Prestiti_Conclusi, Testi, Opere
    WHERE Studenti.IDS = Prestiti_Conclusi.ID_Studente_FK
           AND Prestiti_Conclusi.ID_Testo_FK = Testi.IDT
           AND Testi.ID_Opera_FK = Opere.IDO
    )
)
```




SOMMARIO

CREAZIONE DI UN DATABASE LOCALE CON VISUAL STUDIO.....	2
Creazione di un progetto e di un file di database locale	2
Creazione di tabelle, colonne, chiavi primarie ed esterne	5
Creazione della tabella Studenti.....	5
Creazione della tabella Opere.....	7
Creazione della tabella Testi.....	7
Creazione della tabella Prestiti_Conclusi	7
Per eliminare una tabella sbagliata	7
Per creare una chiave esterna	8
INDICI SUI CAMPI DELLE TABELLE	13
PER VINCOLARE E ANCHE IMPOSTARE DUPLICATI NON AMMESSI	13
Indici, esempio 1	13
INSERIMENTO DI DATI NELLE TABELLE	14
PER INSERIRE DATI NELLE TABELLE	14
CONNESSIONE AI DATI IN UN FILE DI DATABASE LOCALE.....	15
Prerequisiti	15
Creazione e configurazione di un dataset	15
Per creare e configurare un dataset.....	15
VISUALIZZARE DATI CORRELATI IN UN'APPLICAZIONE WINDOWS FORM	19
Creazione di controlli associati a dati per la visualizzazione di record correlati	19
Per creare controlli per la visualizzazione di record correlati	19
DML DI SQL SERVER.....	20
QUERY.....	20
Preparazione di una query su una tabella	20
Preparazione di una query su più tabelle	21
VISTE	22
Vista, Esempio 1	22
Vista, Esempio 2.....	22
FUNZIONI	23
Funzione, Esempio 1	23
Funzione, Esempio 2	24