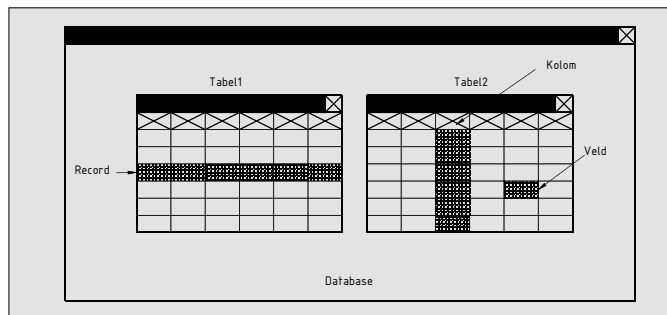


## 17.3

# Database

## Theorie

**G**egevens worden op een gestructureerde wijze opgeslagen in een database. Bij een rekenblad ontbreekt die structuur. De gegevens van een rekenblad kunnen door elkaar raken. Bijvoorbeeld na een sorteeractie. De gegevens van een records van een database blijven wel bij elkaar. Iedere kolom in de database heeft een naam en een type.



U kunt een database bijvoorbeeld gebruiken als tekeningbeheersysteem of voor een stuklijst.

## ADO.NET

ADO.NET is een onderdeel van het .NET Framework. Deze API maakt toegang tot databases mogelijk. Er zijn meerdere providers, met klassen voor de koppeling beschikbaar. Hier worden de twee belangrijkste besproken.

## OLEDB, SQLCe

De koppeling met een database zoals Access vindt plaats via de OLEDB provider. De namespace is `System.data.OleDb`. Als u de database gebruikt die standaard bij .NET wordt geleverd dan gebruikt u `SqlCe`. De namespace is dan `System.data.SqlServerCe`.

*SQL*

De communicatie tussen VB en de database verloopt het makkelijkst via de taal SQL, Structured Query Language. VB stuurt een SQL uitdrukking naar de database en de database stuurt de gegevens terug.

*ConnectionString*

In principe werkt u met iedere database op dezelfde manier. Het enige dat bij iedere database anders is de Connectionstring. Dit is een tekstregel die bepaalt hoe de verbinding gelegd moet worden. De verbindingsregel bestaat meestal uit twee instellingen gescheiden door het teken punt-komma. In deze tekst staat de instelling van de Driver (Driver=) en de bestandsnaam van de database (DBQ=). Voor SQLCe is de database voldoende:

```
verbinding= "Data Source=|DataDirectory|\cursus.sdf"
```

## Praktijk



Microsoft Office  
Access 2007



Lege Database

Cursus

Maken

A

B



Ja

Tabel1

OK

## Aanmaken database

In deze les wordt gewerkt met een Access Database. In het vorige boek over AutoCAD 2011 is de database van Visual Studio gebruikt.

Start Access. Waarschijnlijk staat er op de desktop van uw computer wel dit symbool.

Start met een nieuwe lege database. Het icoon van Access 2007 ziet er iets anders uit dan die van 2010.

Noem de database Cursus.

Maak de database aan.

Type in de tweede kolom de letter A en Enter.

Type in de nieuwe kolom de letter B en Enter.



Access 2007 en Access 2010 nemen aan dat u dan twee nieuwe kolommen wilt aanmaken van het type tekst. De eerste kolom is automatisch in een AutoNummering kolom veranderd.

Sluit Access.

Er wordt gevraagd of u de wijzigingen wilt opslaan. Bevestig dit.

Er wordt gevraagd hoe u de tabel gaat noemen.

Accepteer de standaardnaam Tabell1.

## Praktijk

Project

Add Existing Item

All Files (\*.\*)

Cursus.acdb

Add

Add link

Next

 Id Veld1 Veld2

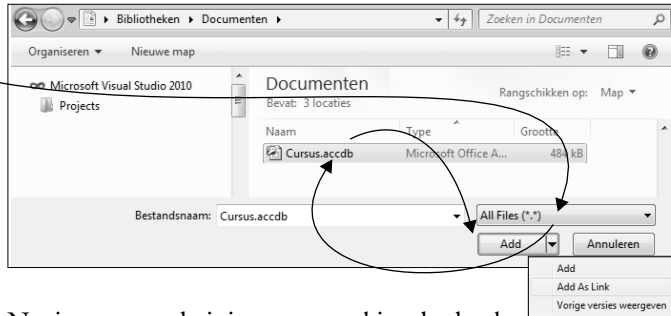
Finish

## Koppelen Database

U werkt verder met Visual Studio.

Activeer de functie om een bestaand item te koppelen. Er verschijnt een bestandskeuze dialoogbox.

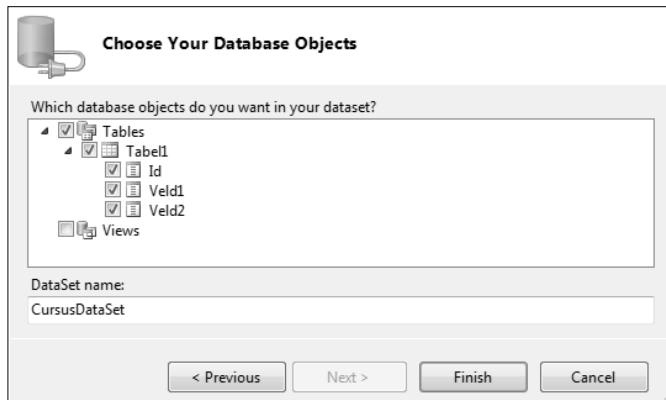
Kies voor alle bestanden.



Navigeer naar de juiste map en kies de database die u hier-voor heeft aangemaakt.

Klik op de pijl naast Add en en daarna op Add Link. Hier-mee kiest u voor een koppeling terwijl het bestand op zijn originele plaats blijft. Er verschijnt een Wizard.

Kies voor de volgende stap.



Kies de velden die u wilt bewerken met Visual Studio. U selecteert de drie kolommen van tabel1.

Accepteer de standaardnaam voor deze verbinding.

## 17.4

# Database commando's

## Theorie

**W**anneer u eenmaal een koppeling heeft gelegd zult u voornamelijk verder werken met twee objecten:

- OleDbConnection (SqlCeConnection)
- OleDbCommand (SqlCeCommand)

Werkt u met Access, dan gebruikt u de variant OleDb, anders gebruikt u SqlCe.

## OleDbConnection,

Bij het instantieren van dit object geeft u de connectiestring.

```
Dim database = New OleDbConnection(ConnectieString)
```

```
Of Dim database = New SqlCeConnection(ConnectieString)
```

Het resultaat is een database object die bewerkt gaat worden of waaruit informatie wordt opgevraagd. De belangrijkste Methoden van dit object zijn:

- Open: Hiermee start u de verbinding.
- BeginTransaction: Start de transactie. Deze bevestigt u met Commit en verwerpt u met Dispose of Rollback.
- Close: Hiermee sluit u de verbinding.

## OleDbCommand

Met dit object raadpleegt/verandert u de database. Bij het instantieren geeft u de volgende gegevens op:

```
Dim SCC = New OleDbCommand(SqlString, Database, [transactie])
```

```
Of Dim SCC=New SqlCeCommand(SqlString, Database, [transactie])
```

Het verkregen object heeft twee belangrijke methoden:

- `ExecuteNonQuery`: Deze wordt gebruikt voor bewerkingen waarbij geen gegevens worden teruggegeven, zoals het toevoegen of het wijzigen van een database.
- `ExecuteReader`: Deze methode wordt gebruikt voor commando's die een resultaat teruggeven, zoals bij het opvragen van informatie uit een database. Het resultaat is een `DataReader`object die de gegevens bevat.

Minder belangrijke methode van `SqlCeCommand` is de methode `Parameters`. Hiermee kunt u op een veilige manier variabelen gebruiken in de `Sql`. Dit doet u vooral om de invoer van een gebruiker gecontroleerd in de `SQL` string te verwerken. Bijvoorbeeld om te voorkomen dat de gebruiker zelf een `SQL` commando geeft bij de invoer.

### *Datareader*

Dit is een readonly object waarin de gegevens zijn opgeslagen die gemaakt zijn met de methode `ExecuteReader`. U kunt deze gegevens niet veranderen. Om een regel uit dit object uit te lezen wordt de methode `Read` uitgevoerd. Als de regel bestaat dan geeft deze methode `True` als resultaat en als de regel niet bestaat, dan `False`. In een lus kunt u alle records doorlopen. De gegevens in de record kunt u benaderen alsof het een array betreft. Index 0 is het eerste veld van de record, index 1 het tweede veld.

## Praktijk

Project
Add Module
SQLDatabase
Add

## Vorbereiding

U werkt verder met het project `Cursus`. Activeer het commando om een nieuwe database aan te maken.

Geef deze module de naam uit de kantlijn.

Controleer de invoer en sluit het venster.

Voeg de namespaces toe en maak de module publiek.

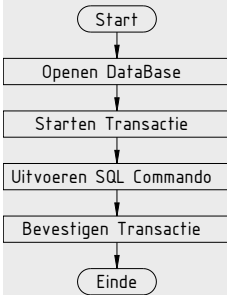
```
Imports System.Data.OleDb

Public Module SQLDatabase

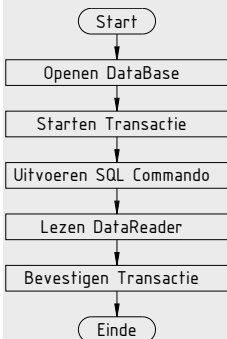
End Module
```

## Praktijk

### Opdracht



### Opdracht



## Database Commando's

Maak de volgende functie aan in Module SQLDatabase download deze [[www.cadcollege.com](http://www.cadcollege.com) .....SQLDatabase]:

```

Public Module SQLDatabase

Function Uitvoeren(ByVal SQLString As String) As Integer
Dim Database = New OleDbConnection( _
    My.Settings.CursusConnectionString)
Database.Open()

Using transactie = Database.BeginTransaction()
Try
    Dim SQLCommando = New OleDbCommand(SQLString, _
        Database, transactie)

    SQLCommando.ExecuteNonQuery()
    transactie.Commit()
Catch ex As Exception
    MsgBox("Er ging iets fout: " & vbCrLf & ex.Message)
End Try
End Using
End Function
  
```

Merk op dat de structuur van het programma bijna volledig overeenkomt met de structuur van het tekenen met vb.

Maak de volgende functie aan in Module SQLDatabase

```

Function Lezen(ByVal SQLString As String) As List(Of Object)
Lezen = New List(Of Object)
Dim Database = New OleDbConnection( _
    My.Settings.CursusConnectionString)
Database.Open()
Using transactie = Database.BeginTransaction()
Try
    Dim SQLCommando = New OleDbCommand(SQLString, _
        Database, transactie)

    Dim resultaat = SQLCommando.ExecuteReader()

    Dim regel As New List(Of Object)
    While resultaat.Read
        For i = 0 To resultaat.FieldCount- 1
            regel.Add(resultaat.GetValue(i))
        Next
        Lezen.Add(regel)
    End While

    resultaat.Close()
    transactie.Commit()
Catch ex As Exception
    MsgBox("Er ging iets fout: " & vbCrLf & ex.Message)
End Try
End Using
End Function
  
```

Merk op dat de structuur van het programma overeenkomt met de structuur van het opvragen van een selectieset met vb. In de volgende paragraaf test u de routines.

## 17.5

# SQL DML

*SQL* **S**QL (Structured Query Language) is een verzameling van instructies waarmee u gegevens kunt ophalen en veranderen. U kunt hiermee ook samenvattingsinformatie opvragen, zoals totalen, gemiddelden. U kunt ermee tabellen maken of verwijderen. Het voordeel van SQL is dat u hiermee met slechts een enkele regel code een functie kan maken.

*DML, DDL* SQL kan opgedeeld worden in twee delen. De Gegevens manipulatietaal (Eng: Data Manipulation Language) en de gegevens definitietaal (Eng: Data Definition Language). Als u een tabel aanmaakt is dat een DDL instructie. Als u gegevens opvraagt is dat een DML instructie.

*Anatomie* Een SQL uitdrukking bestaat uit een aantal delen.

```
Manipulatie instructie [Optiedeclaraties]
```

- Manipulatie instructie; dit deel geeft aan welke actie ondernomen moet worden zoals INSERT of SELECT.
- Optiedeclaraties; deze stellen filter voorwaarden of sorteringen in zoals WHERE of ORDER BY

Een voorbeeld van een SQL uitdrukking is

```
SELECT * FROM Tabell.
```

Met deze uitdrukking worden alle kolommen van Tabell geselecteerd.

*Manipulatie instructie*

De belangrijkste Manipulatiefuncties zijn:

- **SELECT**: Kiest een aantal velden.
- **DELETE FROM**; Verwijdert records uit een tabel.
- **INSERT INTO**: Voegt een groep records toe.
- **UPDATE**: Stelt de waarde van een veld of record in.

*SELECT*

De instructie **SELECT** haalt velden of records op. Deze instructie heeft de volgende syntax.

```
SELECT velden FROM [tabellen] [relaties] [sorteeropties]
```

*Velden*

De velden zijn meestal kolommen uit een tabel. U kunt alle kolommen uit een tabel selecteren met \*. Bijvoorbeeld:

```
SELECT * FROM tabel1.
```

U kunt ook een kolom kiezen zoals `id` of `tabel1.id` als `id` een veld is in de `tabel1`. U kiest twee of meer kolommen door deze te scheiden met een komma, bijvoorbeeld `id, nr` of `tabel1.id, tabel1.nr` als `id` en `nr` kolommen zijn in `tabel1`. Deze kolommen mogen ook uit verschillende tabellen komen. Als een kolomnaam een spatie bevat, dan moet u rechte haken om de kolomnaam plaatsen. De naam van de kolom die u kiest wordt de naam van de kolom in de datareader. Als u dat niet wilt dan gebruikt u de uitdrukking **ALIAS**. Na het woord **ALIAS** staat de nieuwe naam van de kolom. Bijvoorbeeld:

```
SELECT nr ALIAS PosNr FROM tabel1
```

De datareader bestaat dan uit een enkele kolom. De naam van deze kolom in de oorspronkelijke tabel is `nr`. De naam van de kolom in de datareader is `PosNr`.

*Relaties*

Relaties brengt u aan met het sleutelwoord **WHERE**. Achter **WHERE** volgt de conditie. Bijvoorbeeld:

```
WHERE Aantal=2
WHERE Tabel1.Artikel=Tabel2.Artikel.
```

*Sorteren*

Voor het sorteren gebruikt u de sleutel **ORDER BY** gevolgd door de kolomnaam. Als u niet van A tot Z wilt sor-

teren, maar andersom dan voegt u achteraf de sleutel DESC toe. (descendent is aflopend) Bijvoorbeeld:

```
ORDER BY Omschrijving
ORDER BY Omschrijving DESC
```

### Berekeningen

U kunt met SQL ook berekeningen uitvoeren. U behandelt het resultaat alsof u een veld opvraagt. U kunt rekenkundige bewerkingen toepassen (+, -, \*, /). Of statistische: Avg (gemiddeld), Count (aantal), Min, Max, Sum. Ook kunt u wiskundige functies gebruiken: Abs, Atn, Cos, Fix, Int, Round, Sgn, Sqr, Tan. Bijvoorbeeld:

```
SELECT Aantal * Prijs, Omschrijving ...
```

Een tekststring wordt omsloten door een enkel string-teken. Dit is hetzelfde teken dat gebruikt wordt in VB om commentaar aan te geven. U kunt de volgende tekstbewerkingen toepassen (&). U kunt de volgende tekstfuncties gebruiken: Asc, Chr, Chr\$, Format, InStr etc Bijvoorbeeld

```
SELECT Nr & ', ' & Omschrijving
```

### Tabellen

Behalve dat u aangeeft welke informatie u wilt hebben moet u ook aangeven uit welke tabellen die informatie moet komen. Hiervoor gebruikt u FROM:

```
FROM tabel1 [AS alias1][,tabel2[AS alias2]
```

U gebruikt een alias om de naam van de tabel af te korten, zodat u bij het WHERE gedeelte van de SELECT uitdrukking de afkorting kunt gebruiken.

### WHERE

In het gedeelte, waar WHERE staat, wordt aangegeven welke records er geselecteerd gaan worden. U vergelijkt twee of meer waarden uit een of meer tabellen en als de uitdrukking waar is dan worden de betreffende records geselecteerd. Bijvoorbeeld:

```
SELECT omschrijving FROM tabel1,tabel2 WHERE
        tabel1.artikel=tabel2.artikel
```

Stel er zitten vier records in tabel1 en vier records in tabel2. Zonder WHERE zouden zestien records gekozen zijn. (4 \* 4) In slechts 4 van de zestien gevallen is tabel1.artikel gelijk aan tabel2.artikel. Daarom worden er met SELECT slechts

vier records gekozen. U kunt in de vergelijking gebruik maken van: =, <, <=, >=, > en <>.

*INTO*

De SQL uitdrukkingen die u tot nu toe heeft uitgevoerd hebben een datareader aangemaakt. Deze verdwijnt aan het eind van uw procedure als u de methode Close uitvoert. U kunt deze gegevens ook permanent wegschrijven in een tabel met INTO. Bijvoorbeeld:

```
SELECT * INTO test FROM tabel1
```

Tot nu toe heeft u met behulp van de uitdrukking Select meerdere selecties aangemaakt. Er zijn echter veel meer SQL uitdrukkingen:

- DELETE FROM; deze verwijdert een record.
- INSERT INTO; deze voegt een record toe.
- UPDATE; deze verandert een record.

*Delete*

De uitdrukking Delete heeft geen resultaat. De syntax is:

```
DELETE FROM tabelnaam [WHERE clause]
```

Als u geen WHERE clause meegeeft, dan worden alle records verwijderd, zoals bij Select.

*Insert*

Met deze uitdrukking voegt u een groep records toe aan een tabel. De syntax heeft twee vormen:

```
INSERT tabel SELECT...
```

Met deze uitdrukking maakt u een selectie en voegt deze toe aan een andere tabel. U kunt ook een record toevoegen deze direct van een waarde voorzien.

```
INSERT INTO tabelnaam VALUES (kolom1, kolom2...)
```

U dient alle waarden van het record aan te geven. Bijvoorbeeld:

```
INSERT INTO tabel2 VALUES (10000003, 'test', true, 2)
```

*Update*

Met de uitdrukking Update kunt u bepaalde delen van de database wijzigen.

```
UPDATE tabel SET veld=waarde [WHERE ...]
```

## Praktijk

Project

Add Windows Form

SQLTest

Add



DataGridView



(None) ▼

Add Project Data Source

Opdracht

## SQL Commando's

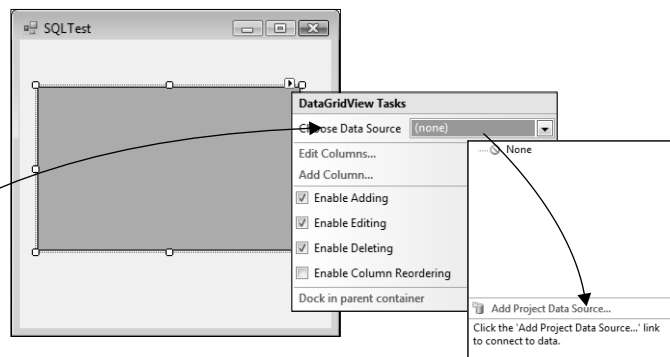
Activeer het commando om een nieuw formulier toe te voegen.

Noem dit formulier SQLTest.

Controleer de invoer en klik op Add.

Klik in de toolbox onder de groep Data op de knop DataGridView.

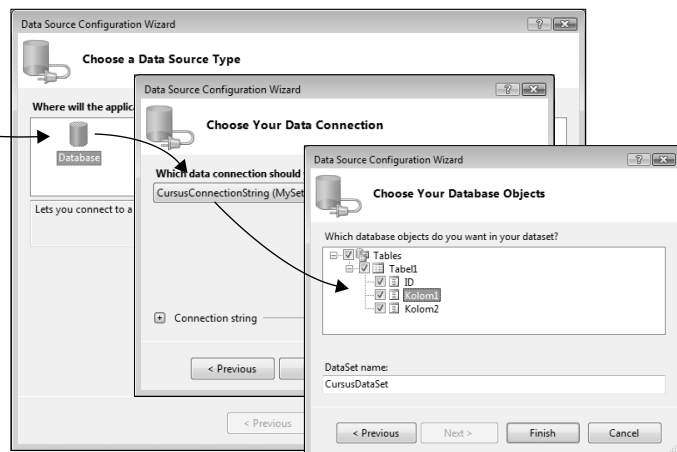
Klik op het formulier. Er verschijnt een control en er verschijnt een menu.



Klik op de lijst met database verbindingen.

Klik op de link om een data verbinding op te zetten.

Gebruik de wizard om alle drie de kolommen van de tabel te kiezen.



Finish

Controleer of alle drie de kolommen zijn geselecteerd en klik op Finish. Het datagrid wordt zichtbaar op de form.

Opdracht

Voeg nog een aantal knoppen toe aan het formulier.



Opdracht

Dubbelklik iedere keer op een knop en vul de code aan. Vul de namespaces aan en maak een testroutine of kopieer deze uit [www.cadcollege.com > .. > SQLTest]:

```
Imports Autodesk.AutoCAD.Runtime
Imports Autodesk.AutoCAD.ApplicationServices
```

```
Public Class SQLTest
```

```
<CommandMethod("Cursus_SQL_Test")> Public Sub LB()
    Dim formulier = New SQLTest
    Application.ShowModelessDialog(formulier)
End Sub
```

```
Private Sub SQLTest_Load( . . . ) Handles MyBase.Load
    Me.TabellTableAdapter.Fill(Me.CursusDataSet.Tabell)
End Sub
```

```
Private Sub cmdInvoegen1A_Click(. . .) . . .
    DIM S="INSERT INTO tabell(veld1,veld2) VALUES ( '1', 'a' )"
    SQLDatabase.Uitvoeren(S)
    Me.TabellTableAdapter.Fill(Me.CursusDataSet.Tabell)
End Sub
```

```
Private Sub cmdInvoegen2B_Click(. . .) . . .
    Dim S="INSERT INTO tabell (veld1,veld2)VALUES ( '2', 'b' )"
    SQLDatabase.Uitvoeren(S)
    Me.TabellTableAdapter.Fill(Me.CursusDataSet.Tabell)
End Sub
```

```
Private Sub cmdVeranderen1C_Click(. . .) . . .
    DIM S = "UPDATE tabell SET veld2 = 'c' WHERE veld1 = '1' "
    SQLDatabase.Uitvoeren()
    Me.TabellTableAdapter.Fill(Me.CursusDataSet.Tabell)
End Sub
```

```
Private Sub cmdToon1_Click(. . .) Handles . . .
    Dim S = "SELECT veld2 FROM tabell WHERE veld1= '1'"
    Dim antwoord = SQLDatabase.Lezen(S)
    MsgBox(antwoord(0)(0))
End Sub
```

```
Private Sub cmdVerwijderAlles_Click(. . .) . . .
    SQLDatabase.Uitvoeren("Delete from tabell")
    Me.TabellTableAdapter.Fill(Me.CursusDataSet.Tabell)
End Sub
```

```
End Class
```

Iedere routine bestaat uit het opbouwen van de SQL-string en het uitvoeren van die string. Alleen de subroutine cmdToon. Deze routine heeft ook nog uitvoer. De uitvoer bestaat in het algemeen uit 1 of meer records met 1 of meer kolommen. Het veld uit de eerste record en eerst kolom is veld (0)(0).

### Opracht

Test de routine uit.

