

SYBASE

TECHWAVE

SYMPOSIUM 2009

---

# Developing Applications for SQL Anywhere Using Visual Studio

Eric Farrar

Product Manager



# Agenda

- SQL Anywhere and Visual Studio
- Database Application Development
  - Desktop
  - Web
  - Mobile
- ADO.NET Entity Framework – *Demonstration*
- CLR External Environment – *Demonstration*
- MobiLink .NET Synchronization Logic – *Demonstration*
- Additional Resources

# SQL Anywhere and Visual Studio

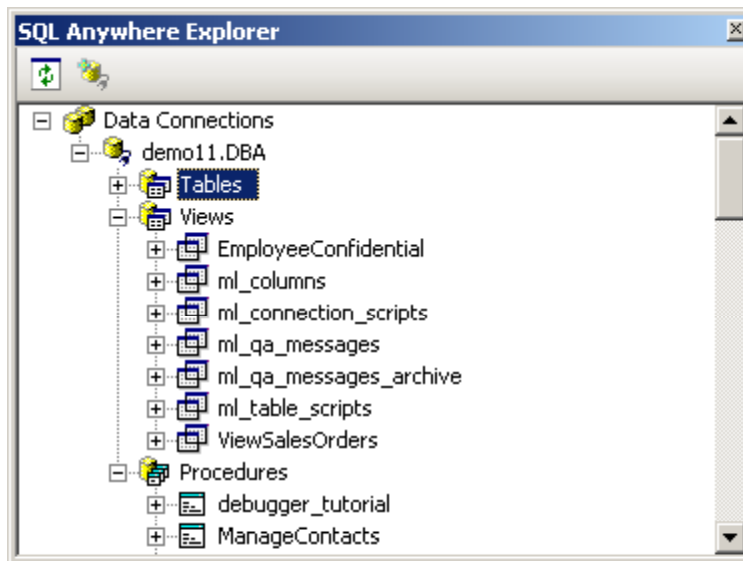
- SQL Anywhere .NET Data Provider
  - .NET Framework 2.0 and 3.x
  - .NET Compact Framework 2.0 and 3.5
- SQL Anywhere Explorer for Visual Studio
  - Visual Studio 2005 and 2008
  - Desktop and Web (ASP.NET) applications
  - Connect to SQL Anywhere and UltraLite databases
- Support for new features in VS 2008, including the Entity Framework

# SQL Anywhere Explorer

- Visual Studio component to view connections from SQL Anywhere and UltraLite databases
- Launch Sybase Central and Interactive SQL
- Drag & drop database objects onto code editor to automatically generate code
  - Data connection → data connection
  - Table → data adapter
  - View → data adapter
  - Stored procedure or function → command
- *SQL Anywhere Explorer functionality will be moved to the Server Explorer in future releases*

# SQL Anywhere Explorer – Demonstration

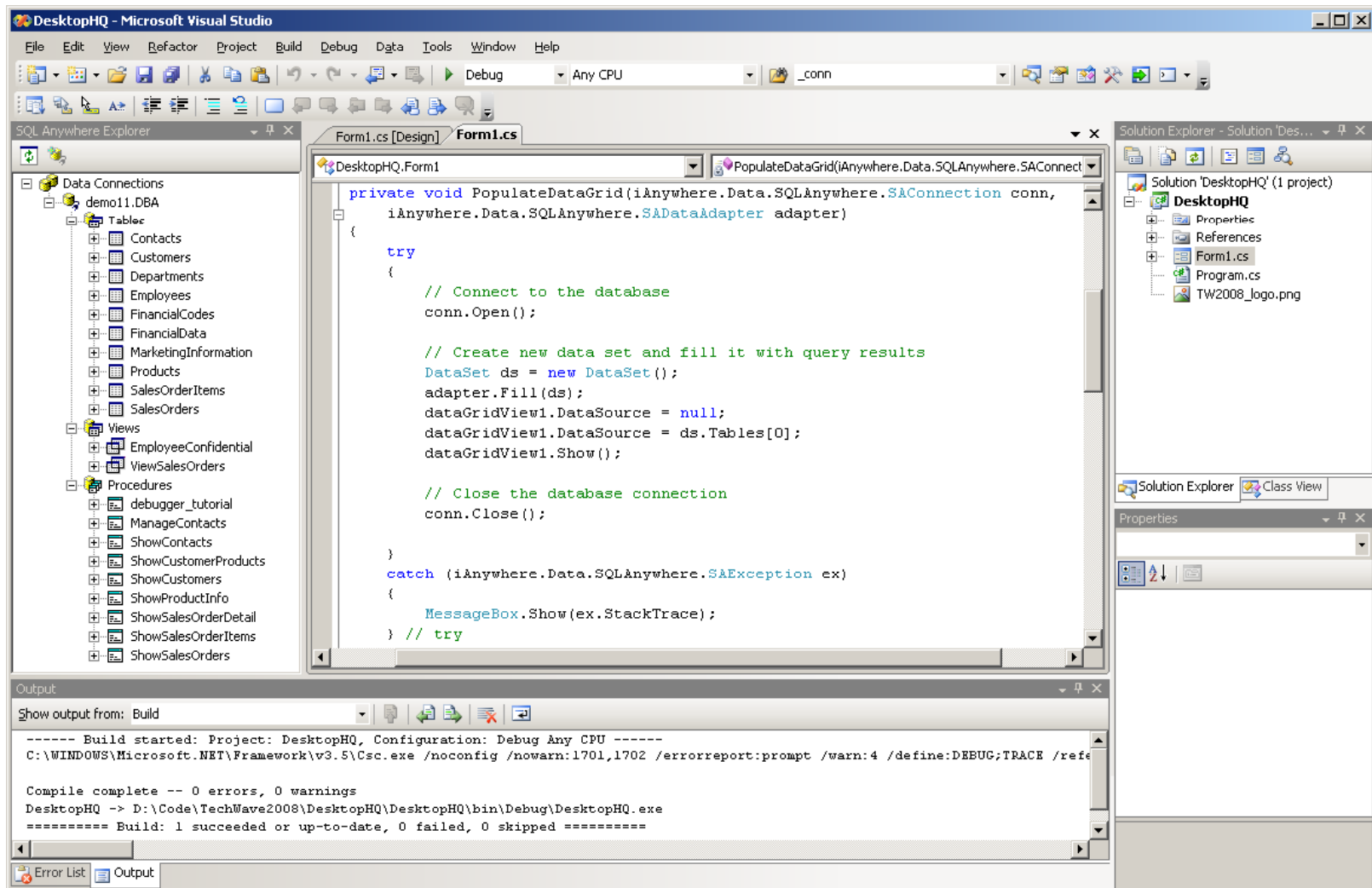
- Adding new connections
- Drag & drop
- Configuring the SQL Anywhere Explorer



# Database Application Development – Desktop

- Types of Database Applications
  - Client/Server
  - Local database with sync
  - Embedded database
- Database Interfaces
  - ADO.NET
  - Entity Framework – new in VS 2008 SP1
- Deployment
  - Database file
  - Database server binaries
  - Client interface (ADO.NET Data Provider)

# Desktop Database Application – Demonstration

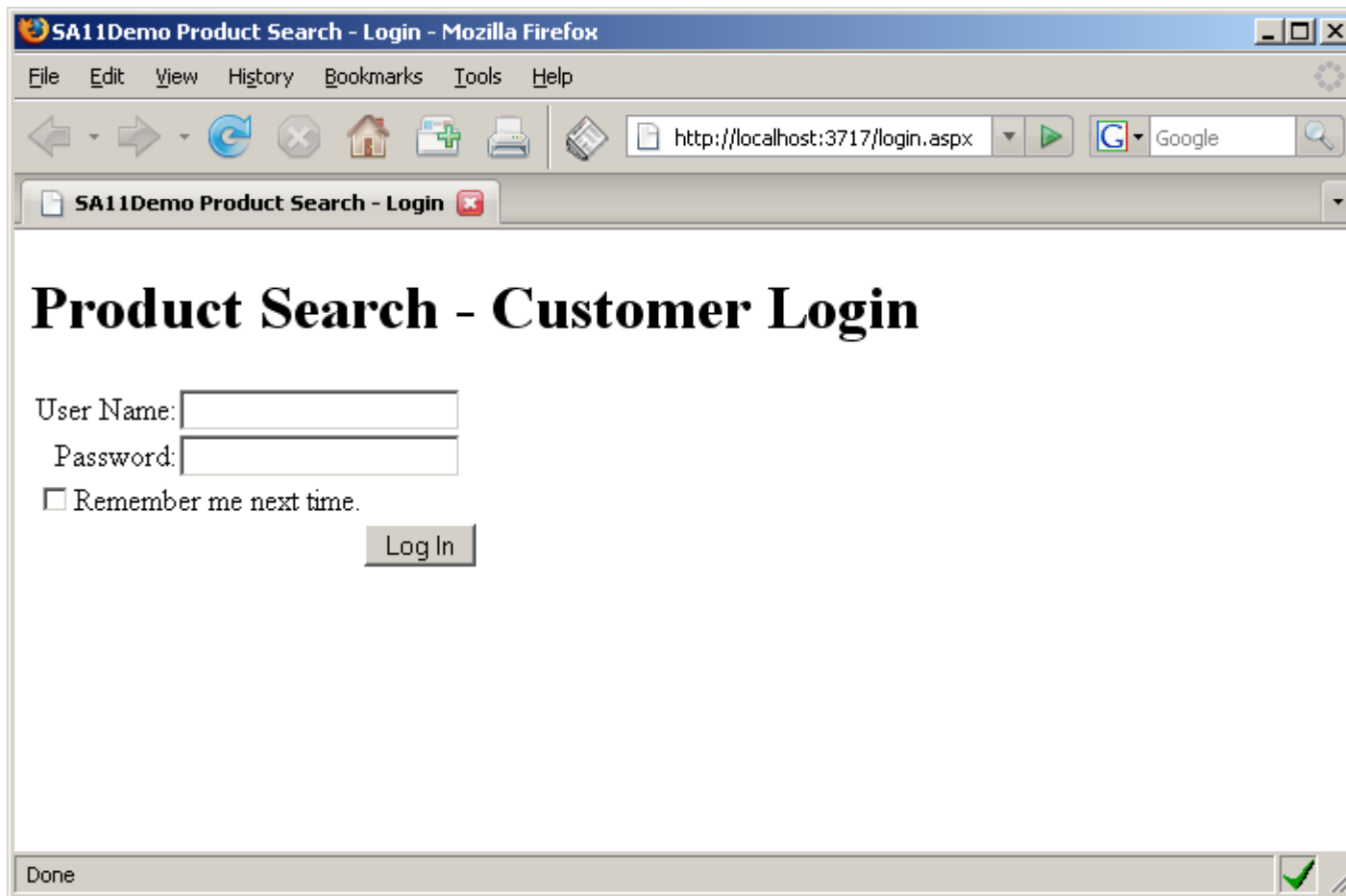


# Database Application Development – Web

- ASP.NET
- Login Credentials
  - SQL Anywhere support for ASP.NET providers (wizard)
  - Implement new Membership Provider class
  - Verify credentials from SQL Anywhere database
- Web Services
  - Produce and consume (handle SOAP requests)
- Deployment
  - Database file
  - Database server binaries
  - Client interface (ADO.NET Data Provider)
  - Make sure IIS can see all components



# Web Database Application – Demonstration



The screenshot shows a Mozilla Firefox browser window with the following details:

- Title Bar:** SA11Demo Product Search - Login - Mozilla Firefox
- Menu Bar:** File Edit View History Bookmarks Tools Help
- Address Bar:** http://localhost:3717/login.aspx
- Search Bar:** Google
- Tab:** SA11Demo Product Search - Login
- Page Content:**
  - Header:** Product Search - Customer Login
  - Form Fields:**
    - User Name:
    - Password:
    - Remember me next time.
  - Button:** Log In
- Status Bar:** Done

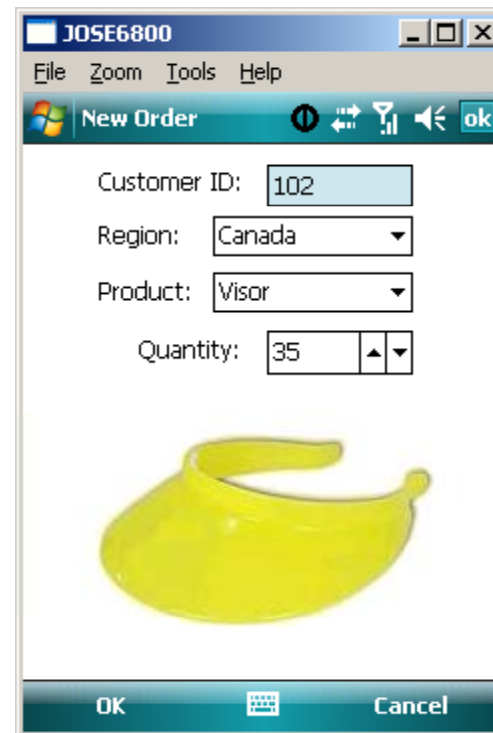
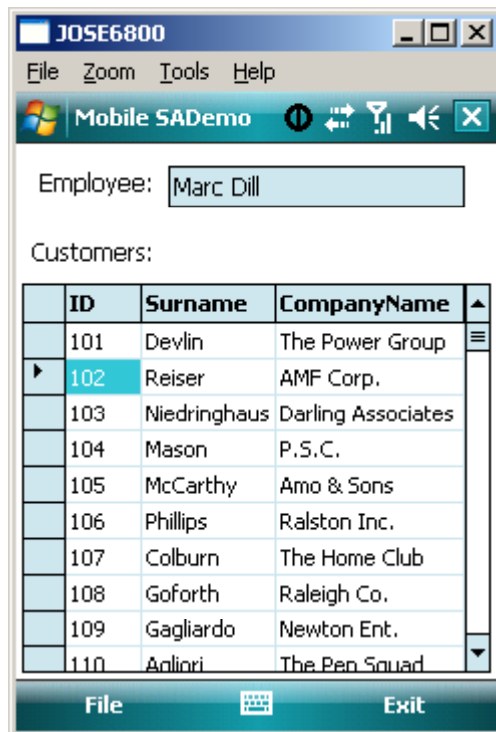
# Desktop & Web Database Applications – Considerations

- Add reference to iAnywhere.Data.SQLAnywhere.dll
- ADO.NET 2.0 or 3.x
- SQL Anywhere Explorer
  - Connect to database from Visual Studio
  - Drag & drop into code editor (not form!)
- If you want to drag & drop onto form, use Server Explorer

# Database Application Development – Mobile

- Windows Mobile
  - SQL Anywhere (ADO.NET)
  - UltraLite (UltraLite.NET)
- Security
  - Device theft or loss
  - Data interception
- Synchronization
  - Use MobiLink Synchronization Wizards
- Deployment
  - Database file
  - Database server binaries
  - Client interface (ADO.NET Data Provider) **only** if using SQL Anywhere

# Mobile Database Application – Demonstration



# Mobile Database Applications – Considerations

- References
  - SQL Anywhere → `iAnywhere.Data.SQLAnywhere.dll`
  - UltraLite → `iAnywhere.Data.UltraLite` and `iAnywhere.Data.UltraLite.resources`
  - Ensure to use DLLs from correct folder (CE vs. Win32)
- SQL Anywhere Explorer for automatic code generation
- UltraLite
  - Must deploy runtime DLL
  - Be careful when adding the UL database to Visual Studio project
    - May break synchronization if constantly deploying same copy of udb

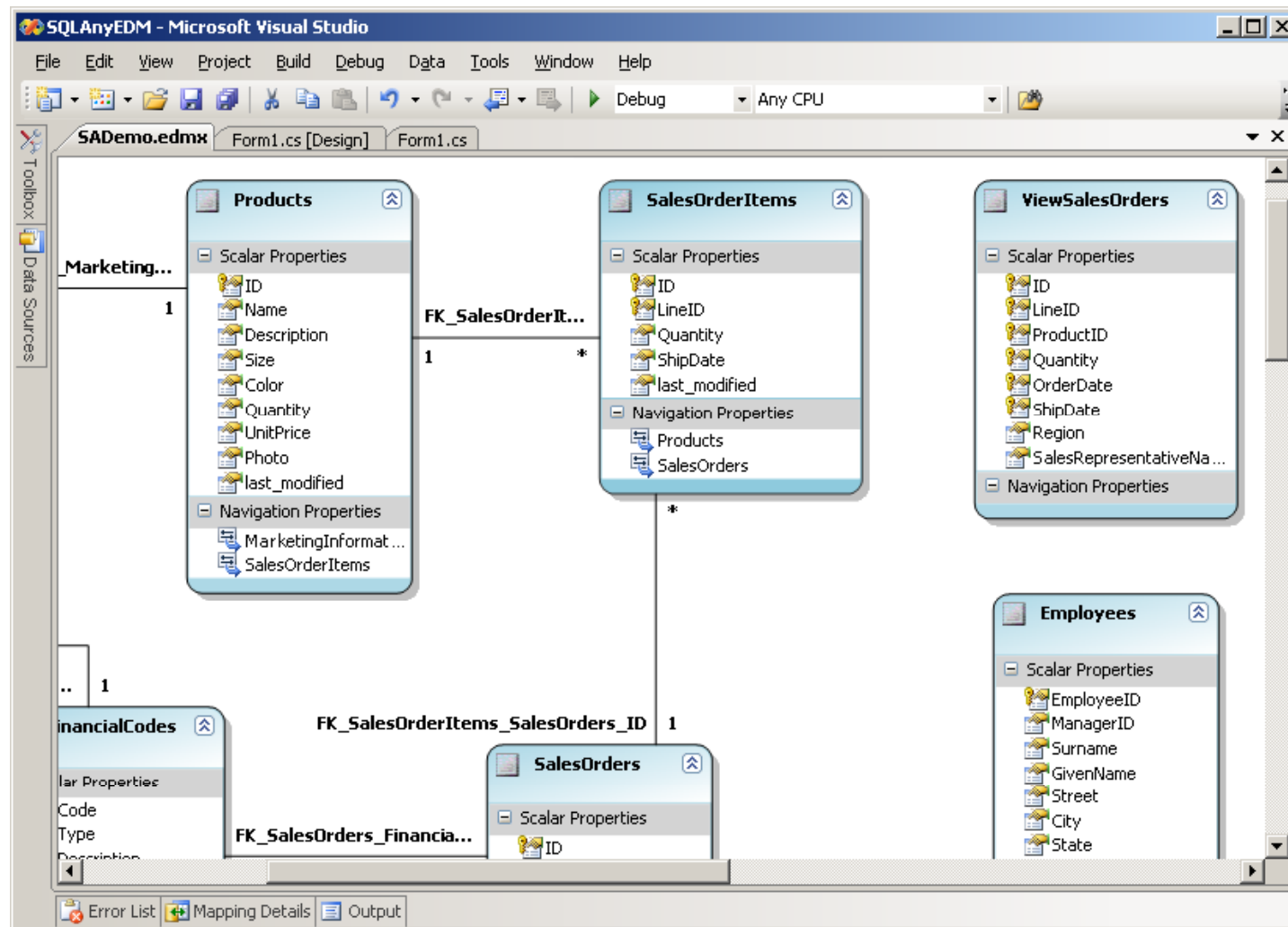
# ADO.NET Entity Framework Support

- SQL Anywhere 11 supports ADO.NET Entity Framework
- Create entity models from SQL Anywhere data sources
- Access data using...
  - Control data binding
  - EntityClient provider
  - Object services
  - LINQ to entities

# Entity Framework Requirements

- Visual Studio 2008 SP1
- .NET Framework 3.5 SP1
  
- SQL Anywhere 11.0.0.1390 or greater

# ADO.NET Entity Framework – Demonstration





# CLR External Environment

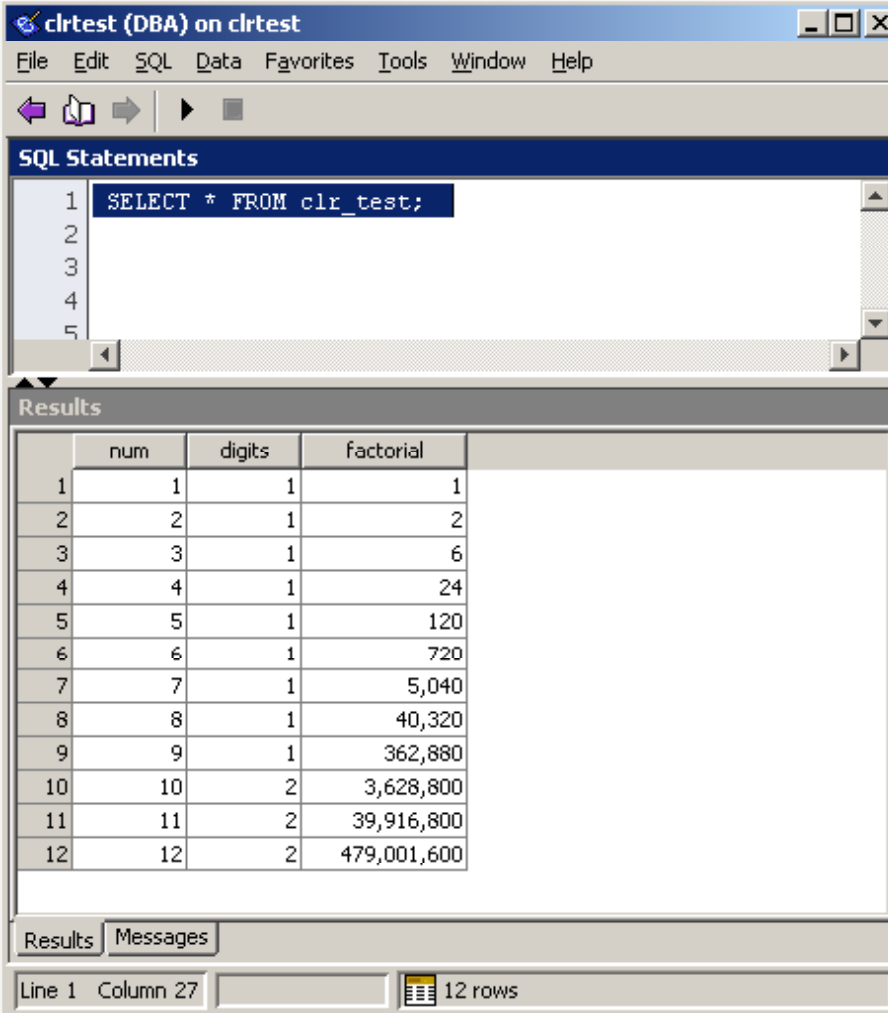
- New feature in SQL Anywhere 11
- Write stored procedures or functions in C# or VB.NET
- Execution takes place outside of the database server
- Support for .NET Framework 2.0, 3.0, and 3.5
- Example:

```
CREATE PROCEDURE clr_stored_proc(  
    IN p1 INT,  
    IN p2 UNSIGNED SMALLINT,  
    OUT p3 LONG VARCHAR)  
EXTERNAL NAME 'MyCLRTest.dll::MyCLRTest.Run( int, ushort, out string )'  
LANGUAGE CLR;
```

# CLR External Environment (cont.)

- Database server uses CLR executable: *dbextclr11.exe*
- CLR executable *does not* search GAC for assembly
- No need to execute INSTALL statement
- Benefits:
  - Leverage existing knowledge of .NET programming languages
  - Re-use existing libraries
  - .NET language better suited than SQL for certain operations (e.g. string manipulation, math computations)

# CLR External Environment – Demonstration



The screenshot shows a SQL client window titled "clrtest (DBA) on clrtest". The window has a menu bar with "File", "Edit", "SQL", "Data", "Favorites", "Tools", "Window", and "Help". Below the menu bar is a toolbar with navigation icons. The "SQL Statements" pane contains the following query:

```
1 SELECT * FROM clr_test;  
2  
3  
4  
5
```

The "Results" pane displays a table with the following data:

	num	digits	factorial
1	1	1	1
2	2	1	2
3	3	1	6
4	4	1	24
5	5	1	120
6	6	1	720
7	7	1	5,040
8	8	1	40,320
9	9	1	362,880
10	10	2	3,628,800
11	11	2	39,916,800
12	12	2	479,001,600

At the bottom of the window, there are tabs for "Results" and "Messages". The status bar at the bottom indicates "Line 1 Column 27" and "12 rows".

# MobiLink .NET Synchronization Logic

- Write sync scripts in C# or VB.NET instead of SQL
  - MobiLink Server API
  - .NET language may be more appropriate than SQL
  - .NET method returns string to MobiLink
  - Handle authentication using .NET
  - Debug .NET scripts in Visual Studio
- Direct row handling
  - Synchronize to non-relational data sources
  - Data source must have unique primary keys
- Portability across different RDBMS types

# MobiLink .NET Sync Logic – Demonstration

The screenshot shows the Microsoft Visual Studio IDE in a debugging state. The main window displays the code for `MExample.CustdbScripts`. The `DownloadCursor` method is selected, and its implementation is visible. The code includes an `INSERT` statement, a `MessageBox.Show` call, and a `SELECT` statement. The `Locals` window at the bottom shows the values of the `ts` and `user` variables.

```
return ("INSERT INTO ULCustomer(cust_id,cust_name) values (?,?)");

lic static string DownloadCursor(System.DateTime ts, string user)
{
    MessageBox.Show("DownloadCursor for user " + user, "MExample");

    return (@"SELECT cust_id, cust_name
            FROM ULCustomer
            WHERE last_modified >= ' " + ts.ToString("yyyy-MM-dd hh:mm:ss.fff");
```

Name	Value	Type
ts	{8/1/2008 11:08:49 AM}	System.D
user	"50"	string

# Summary

- SQL Anywhere has supported .NET since this technology was introduced in 2002
- SQL Anywhere 11 continues to support new .NET data access APIs, such as the ADO.NET Entity Framework
- Developers can take advantage of .NET capabilities when creating stored procedures and functions

# Additional Resources

- [Sybase.com/ianywhere](http://Sybase.com/ianywhere)
  - Sample .NET code
  - Whitepapers and tutorials about ADO.NET and the Entity Framework
  - DOT.NET developer center
- Product documentation
  - Lots of samples
  - Available online with user comments

Questions?



*Thank you!*