



SQL Server 2005

Shlomy Gantz

President, BlueBrick Inc.



Agenda

- SQL Server 2005 overview
- SQL Management Studio
- T-SQL Enhancements
- XML
- Additional Enhancements



Evolution of SQL Server

- 1993 – SQL Server 4.2
- 1995 – SQL Server 6.5
- 1998 – SQL Server 7.0
- 2000 – SQL Server 2000

- 2005 – SQL Server 2005



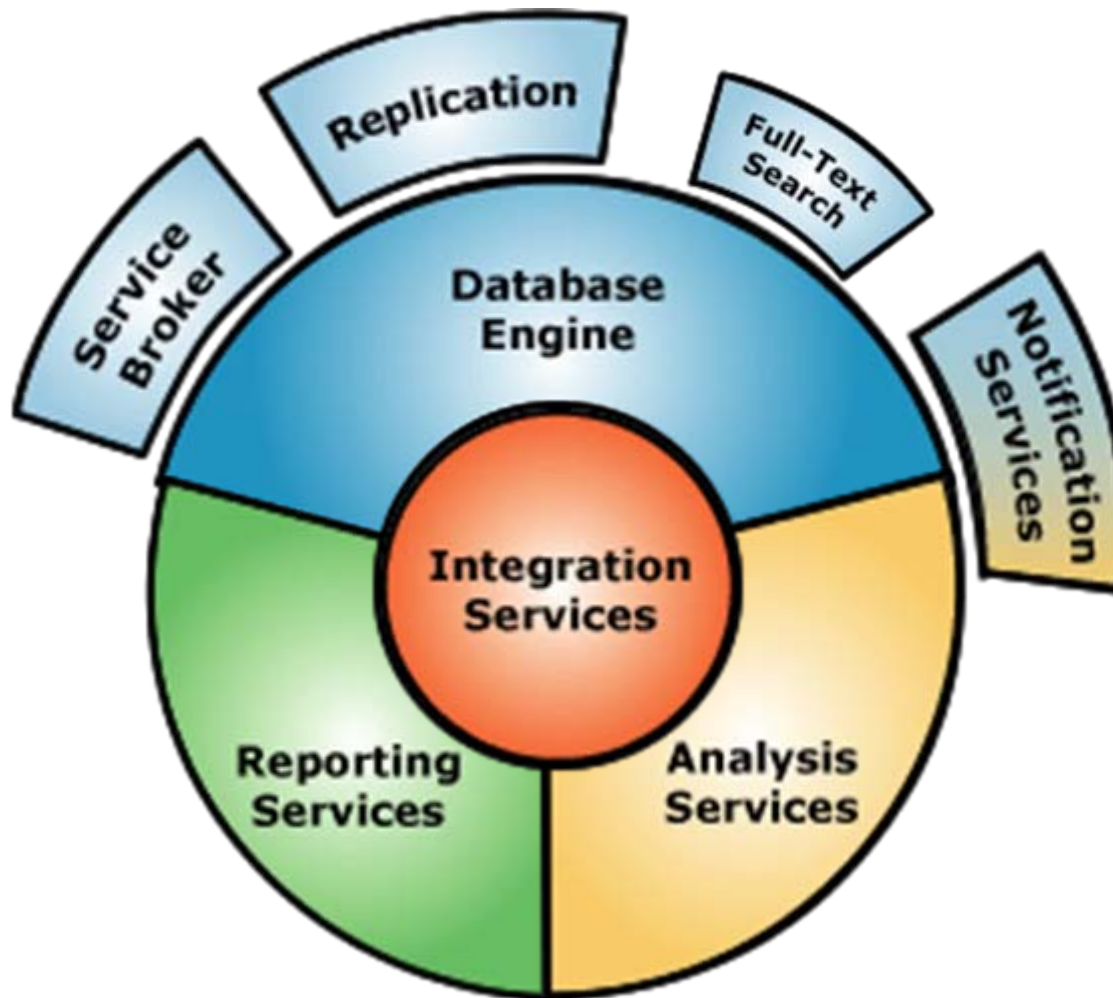
SQL Server 2005 Editions

- Enterprise
- Developer
- Standard
- Workgroup
- Express

SQL Server Express

- Scaled down and easy to use version of SQL 2005
 - CPU's = 1
 - Memory = 1 GB
 - Database size = 4 GB
 - Users = unlimited
 - Cost = FREE
- Replaces SQL Server 2000 MSDE
- Redistributed version of SQL Server for client applications
- Intended for ISVs, ISPs, ASPs, web developers and hobbyists

SQL Server Overview





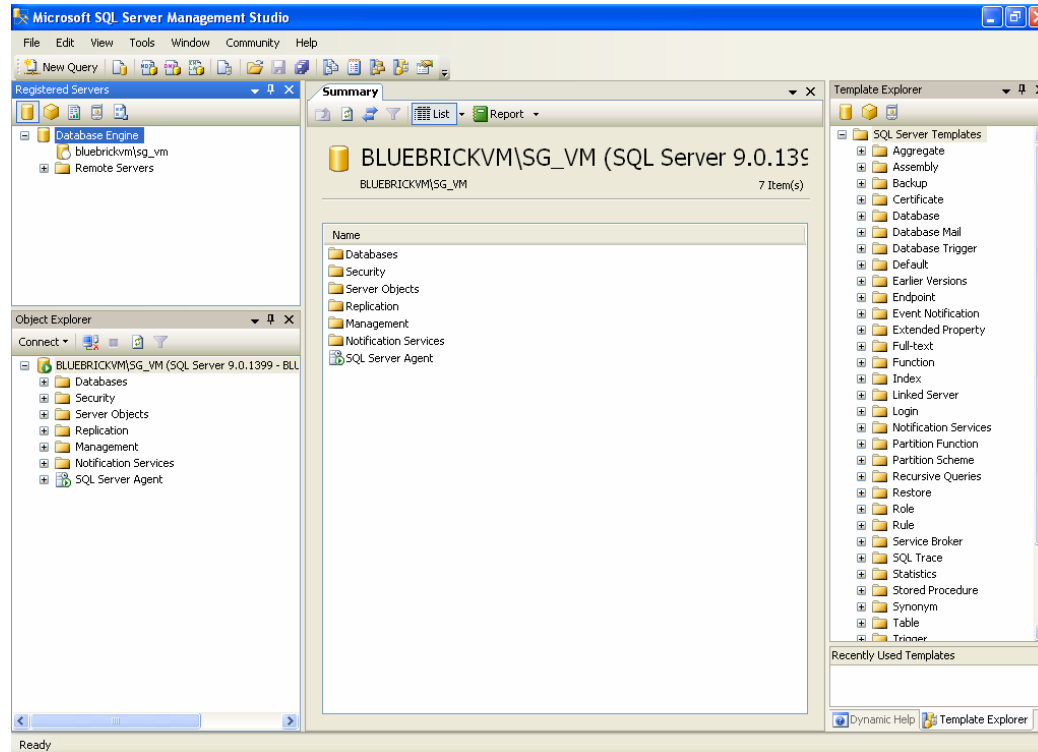
SQL Management Studio



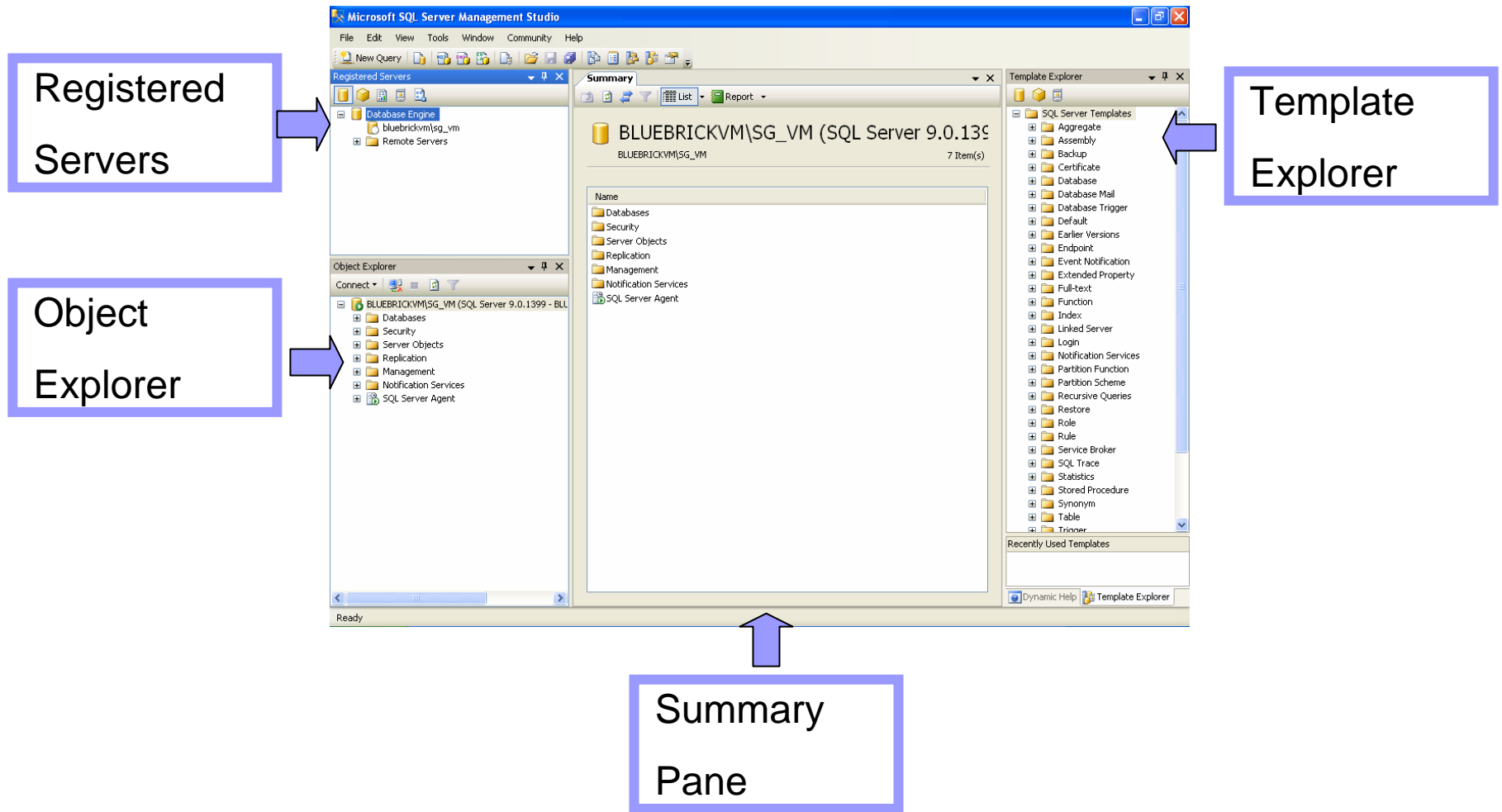
SQL Management Studio

- Brand new interface based on VS
- Supports previous versions
- A single integrated environment for managing all aspects of databases and services

SQL Management Studio



SQL Management Studio





SQL Management Studio

- Asynchronous Treeview and Object Filtering
- Nonmodal/Resizable Dialog boxes
- Scriptable output from Dialog boxes



SQL Management Studio

- Source control
- Template explorer
- Autosave
- Deadlock visualization
- Performance monitor correlation
- Server registration Import/Export
- Print from Results Pane
- XML opens to new window



T-SQL Enhancements

T-SQL - CTE

- CTE – Common Table Expressions
 - Simplified SQL statements
 - Recursive SQL
- New WITH statement
 - Once defined results can be used just like a “view” for the duration of the query.
- You can define Multiple CTEs in one query (separate by commas)

T-SQL -CTE

■ WITH Syntax

```
WITH SalesInJan AS (  
  SELECT * FROM SalesbyMonth  
  WHERE month = 'jan')  
  
SELECT * FROM SalesInJan
```

T-SQL -CTE

■ Recursive Queries

```
WITH pbbActionHeirarchy AS (  
  SELECT pbbActions.action_id,pbbActions.parentid  
  FROM pbbActions  
  WHERE PARENTID =1  
  UNION ALL  
  SELECT pbbActions.action_id,pbbActions.parentid  
  FROM pbbActions  
  INNER JOIN pbbActionHeirarchy ON pbbActions.parentID =  
  pbbActionHeirarchy.action_ID  
)  
  
SELECT * FROM pbbActionHeirarchy
```

T-SQL Non ANSI SQL Joins

- *= and =* are no longer supported
- Only affects outer joins

```
SELECT * FROM  
pbbContacts, pbbContacts2Groups, pbbGroups  
WHERE  
pbbContacts.Contact_ID *= pbbContacts2Groups.ContactID  
AND  
pbbContacts2Groups.GroupID =* pbbGroups.Group_ID
```

T-SQL TOP

- TOP no longer requires a literal and works with INSERT, UPDATE, DELETE

```
INSERT TOP (5) INTO topProducts  
SELECT * FROM Products
```

- CAVEAT – There is no way to determine which rows will be inserted..

```
INSERT INTO topProducts  
SELECT TOP (5) * FROM Products  
ORDER BY PTotalSales DESC
```

T-SQL CROSS/OUTER APPLY

- Used for Derived tables and User Defined Functions
- CROSS APPLY – Only returns rowset if the right table source returns data from the parameter value from the left table source
- OUTER APPLY – Returns at least one row from the left , even if no rowset is returned from the right

T-SQL CROSS/OUTER APPLY

```
SELECT PNAME, productCount.TotalSold FROM PRODUCTS
CROSS APPLY (
  SELECT COUNT(ProductID) AS TotalSold FROM Orders
  WHERE PRODUCTS.Product_ID = Orders.ProductID
) AS productCount
```

T-SQL Enhancements

- PIVOT – Take a set of rows and pivot them into columns

```
SELECT Year, [Jan], [Feb], [Mar], [Apr], [May], [Jun],  
[Jul], [Aug], [Sep], [Oct], [Nov], [Dec]  
FROM (  
SELECT year, amount, month  
FROM salesByMonth ) AS salesByMonth  
PIVOT ( SUM(amount) FOR month IN  
([Jan], [Feb], [Mar], [Apr], [May], [Jun],  
[Jul], [Aug], [Sep], [Oct], [Nov], [Dec])  
) AS ourPivot  
ORDER BY Year
```

T-SQL Enhancements

- UNPIVOT - Take a set of columns and unpivot them into rows

```
SELECT Year, cast(Month as char(3)) as Month, Amount
FROM salesByYear
UNPIVOT (Amount FOR Month IN
([Jan],[Feb],[Mar],[Apr],[May],[Jun],
[Jul],[Aug],[Sep],[Oct],[Nov],[Dec])) as unPivoted
```

T-SQL Ranking Functions

- **ROW_NUMBER**: Returns the row number of a row in a resultset.
- **RANK**: Based on some chosen order of a given set of columns, gives the position of the row. It will leave gaps if there are any ties for values. For example, there might be two values in first place, and then the next value would be third place.
- **DENSE_RANK**: Same as RANK but does not leave gaps in the sequence. Whereas RANK might order values 1,2,2,4,4,6,6, DENSE_RANK would order them 1,2,2,3,3,4,4.
- **NTILE**: Used to partition the ranks into a number of sections; for example, if you have a table with 100 values, you might use NTILE(2) to number the first 50 as “1”, and the last 50 as “2”.

T-SQL Enhancements

■ Error Handling – Try Catch

```
BEGIN TRY
    T-SQL Code...
END TRY
BEGIN CATCH
    T-SQL Code...

    ERROR_NUMBER( )
    ERROR_SEVERITY( )
    ERROR_STATE( )
    ERROR_PROCEDURE( )
    ERROR_LINE( )
    ERROR_MESSAGE( )
END CATCH
```



T-SQL – Additional enhancements

- TABLESAMPLE
- OUTPUT
- EXCEPT and INTERSECT

▪

▪

▪



SQL Server 2005 XML

XML – SQL Server 2000

- OPENXML – rowset view over an XML document
- FOR XML – output row data into XML
- updateGrams
- XML BulkLoad

XML – SQL Server 2005

- FOR XML PATH MODE – Specify hierarchy using XPath
- FOR XML TYPE directive – Returns XML as SQL XML datatype
- Inline XSD using XMLSCHEMA



XML – SQL Server 2005

- XML as Native Datatype
- XQUERY
- XML Indexes

XML Datatype

- XML is now a native datatype instead of storing XML in a string
- Both typed (XSD) and untyped are supported (CONTENT)
- Constraints are supported (CHECK)

XML Datatype limitations

- Cannot be PK or FK
- Up to 2GB in size
- 128 Levels of hierarchy
- Cannot be compared, sorted, grouped by
- Cannot be part of a view
- Cannot be converted to *text* or *ntext*

XQUERY

- FLWOR – For, Let, Where, Order By, Return
 - Query() - Returns the XML that matches your query
 - Value() - Returns a scalar value from your XML
 - Exist() - Checks for the existence of XML
 - Nodes() - Returns a rowset representation of your XML

 - Modify() (SQL specific)

XML Indexing

- Created on XML column
 - Primary – 1 row per node (element, attribute, text) to improve speed to the node
 - Secondary – Path, Value, Property
- Table must have a clustered index
- Cannot modify PK of table after XML index is created



Additional Enhancements

SQLCMD

■ SQLCMD

- Command line interface for any version of SQL Server 2005
- Ability to perform any development or administrative function
- Dedicated Administrator Connection (DAC)
- Default location = C:\Program Files\Microsoft SQL Server\90\Tools\binn
- More information - SQLCMD /?

SQL 2005 Security

- No more blank 'sa' password
- Encryption
 - Password
 - Asymmetric key
 - Symetric key
 - Certificate



SQL 2005 Integration Services

- Formerly DTS
- Brand New IDE
- FTP out
- Send Mail uses SMTP (No more MAPI)
- SSIS is saved as XML

.NET Integration

- Rich choice of modern languages
 - VB.Net, C#, MC++, COBOL, Delphi, ...
 - T-SQL continues as a 1st class citizen
- Leverage extensive frameworks
 - .NET framework: Use extensive libraries built by Microsoft
 - Enable 3rd parties to write libraries & extend DB
- Leverage extensive tools support for .NET
 - VS.NET, Borland & 3rd party tools (e.g. profilers)
 - SQL Server Management Studio
- No more need for XPs!



Additional Enhancements

- Systables are now views
- Synonyms
- Varchar(max) nvarchar(max)
varbinary(max)
- DDL Triggers
- Online indexing
- Snapshots



Additional Enhancements

- Database mirroring / Failover clustering
- Native web services (without IIS)
- Indexes on views (enterprise)
- Oracle replication (enterprise)



Thank you

shlomy@bluebrick.com

<http://www.shlomygantz.com/blog>

<http://www.bluebrick.com>