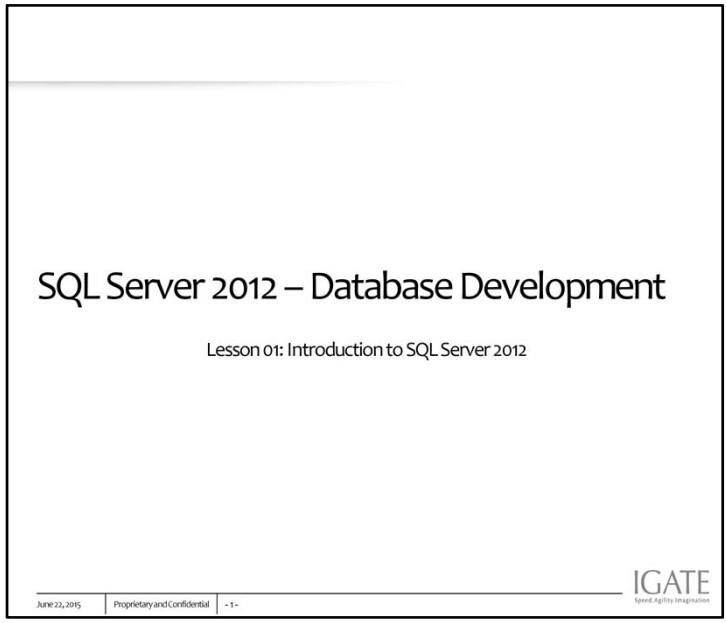
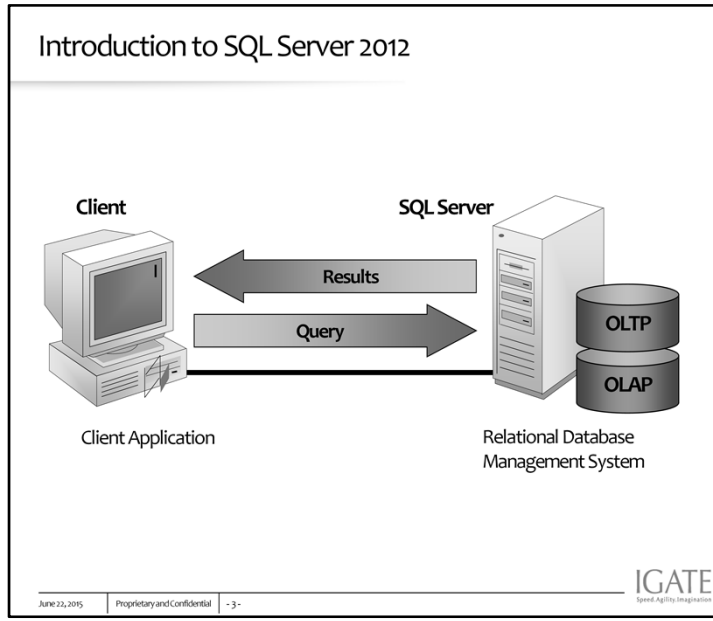

SQL Server 2012 – Database Development
Lesson 01: Introduction to SQL Server 2012
June 22, 2015
Proprietary and Confidential
- 1 -
IGATE
Speed. Agility. Imagination

Lesson Objectives

- **In this lesson, you will learn:**
  - What Is SQL Server?
  - SQL Server Components
  - SQL Server Services
  - SQL Server Key Business Scenarios
  - SQL Server Versions, Editions and Tools
  - Exploring new features in SSMS 2008 and SSMS 2012
  - SQL Server Databases
  - Features of SQL Server



What is SQL Server 2008?

SQL Server 2008 is a database that follows the relational model of data management system.

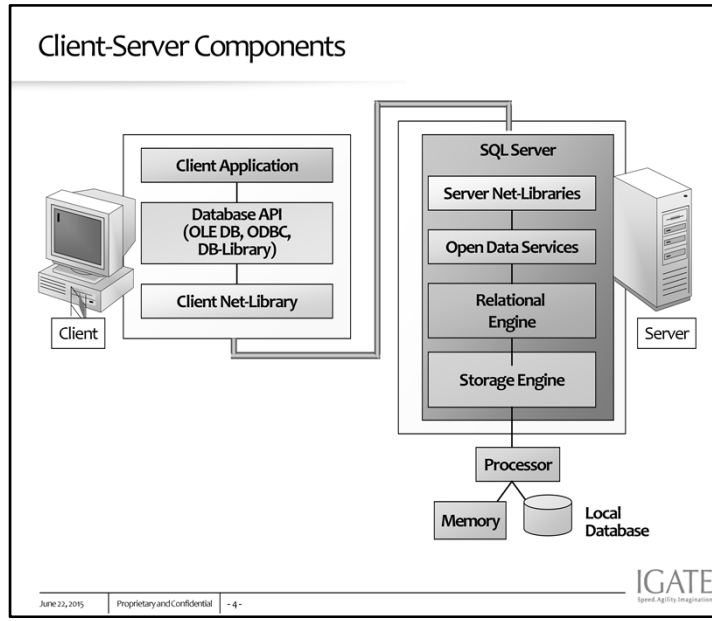
The emphasizes of relational Model is that the data is fundamentally organized in table which can also be termed as a collection of rows and columns .The relational concept maintains the relationships amongst data by adhering to the rules defined by the database administrator. In situations of power failure, which is normally termed as crash situations, the data should be free of corruption (invalid).

SQL Server 2008 is a Relational Database Management System (RDBMS) that provides:

• Maintaining Relationship among Data stored in the Database

• Ensuring Data is stored correctly and rules defining relationship are not violated.

• Recovering all data to a point of consistency , in the event of failure.



The Net-Library

The Net-Library abstraction layer enables SQL Server to read from and write to many different network protocols, and each such protocol (such as TCP/IP sockets) can have a specific driver. The Net-Library layer makes it relatively easy to support many different network protocols without having to change the core server code.

Open Data Services

Open Data Services (ODS) functions as the client manager for SQL Server; it is basically an interface between server Net-Libraries and server-based applications, including SQL Server. ODS manages the network: it listens for new connections, cleans up failed connections, acknowledges "attentions" (cancellations of commands), coordinates threading services to SQL Server, and returns result sets, messages, and status back to the client.

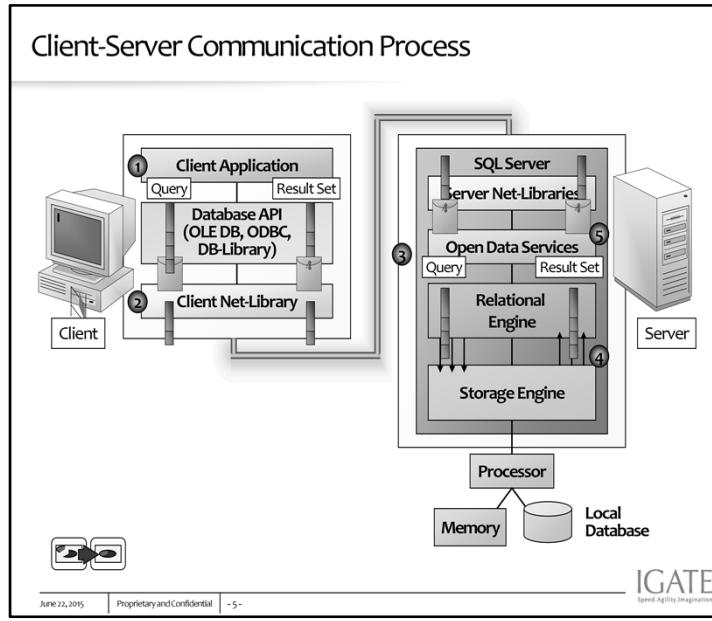


Figure shows the path from the SQL Server client application to the SQL Server engine and shows where the Net-Library interface fits in. On the server side, ODS provides functionality that mirrors that of ODBC, OLE DB, or DB-Library at the client. Calls exist for an ODS server application to describe and send result sets, to convert values between datatypes, to assume the security context associated with the specific connection being managed, and to raise errors and messages to the client application.

ODS uses an event-driven programming model. Requests from servers and clients trigger events that your server application must respond to. Using the ODS API, you create a custom routine, called an *event handler,* for each possible type of event. Essentially, the ODS library drives a server application by calling its custom event handlers in response to incoming requests.

ODS server applications respond to the following events:

Connect events When a connect event occurs, SQL Server initiates a security check to determine whether a connection is allowed. Other ODS applications, such as a gateway to DB/2, have their own logon handlers that determine whether connections are allowed. Events also exist that close a connection, allowing the proper connection cleanup to occur.

Language events When a client sends a command string, such as an SQL statement, SQL Server passes this command along to the command parser. A different ODS application, such as a gateway, would install its own handler that accepts and is responsible for execution of the command.

Remote stored procedure events These events occur each time a client or SQL Server directs a remote stored procedure request to ODS for processing.

The Relational Engine and the Storage Engine

The SQL Server database engine is made up of two main components, the relational engine and the storage engine. Unlike in earlier versions of SQL Server, these two pieces are clearly separated, and their primary method of communication with each other is through OLE DB. The relational engine comprises all the components necessary to parse and optimize any query. It requests data from the storage engine in terms of OLE DB rowsets and then processes the rowsets returned. The storage engine comprises the components needed to actually access and modify data on disk.

Communication Between the Relational Engine and the Storage Engine

The relational engine uses OLE DB for most of its communication with the storage engine. The following description of that communication is adapted from the section titled "Database Server" in the SQL Server Books Online. It describes how a SELECT statement that processes data from local tables only is processed:

The relational engine compiles the **SELECT** statement into an optimized execution plan. The execution plan defines a series of operations against simple rowsets from the individual tables or indexes referenced in the **SELECT** statement. (Rowset is the OLE DB term for a result set.) The rowsets requested by the relational engine return the amount of data needed from a table or index to perform one of the operations used to build the **SELECT** result set. For example, this **SELECT** statement requires a table scan if it references a table with no indexes:

```
SELECT * FROM ScanTable
```

The relational engine implements the table scan by requesting one rowset containing all the rows from ScanTable. This next SELECT statement needs only information available in an index:

```
SELECT DISTINCT LastName
FROM Northwind.dbo.Employees
```

The relational engine implements the index scan by requesting one rowset containing the leaf rows from the index that was built on the LastName column.

The following SELECT statement needs information from two indexes:

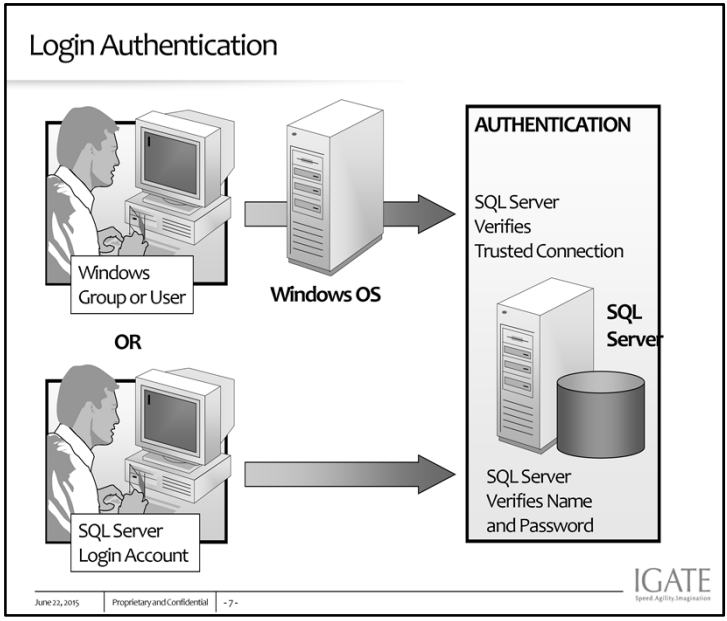
```
SELECT CompanyName, OrderID, ShippedDate
FROM Northwind.dbo.Customers AS Cst
JOIN Northwind.dbo.Orders AS Ord
ON (Cst.CustomerID = Ord.CustomerID)
```

The relational engine requests two rowsets: one for the clustered index on Customers and the other for one of the nonclustered indexes on Orders.

The relational engine then uses the OLE DB API to request that the storage engine open the rowsets.

As the relational engine works through the steps of the execution plan and needs data, it uses OLE DB to fetch the individual rows from the rowsets it requested the storage engine to open. The storage engine transfers the data from the data buffers to the relational engine.

The relational engine combines the data from the storage engine rowsets into the final result set transmitted back to the user.

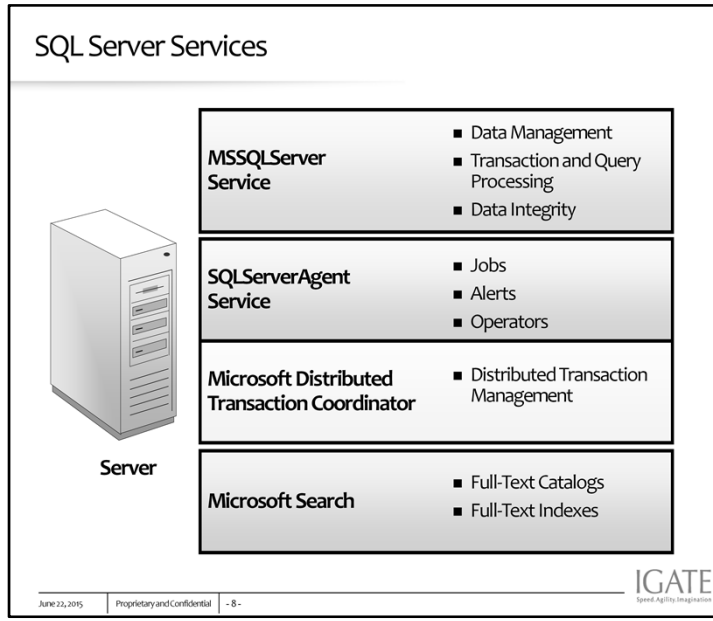


Authentication Modes

Microsoft® SQL Server can operate in one of two security (authentication) modes:

Windows Authentication Mode (Windows Authentication) Windows Authentication mode allows a user to connect through a Microsoft Windows OS user account.

Mixed Mode (Windows Authentication and SQL Server Authentication) Mixed Mode allows users to connect to an instance of SQL Server using either Windows Authentication or SQL Server Authentication. Users who connect through a Windows user account can make use of trusted connections in either Windows Authentication Mode or Mixed Mode.



SQL Server Services
Services are internal processes, which SQL Server invokes to perform specific actions. The major four services of SQL-Server are as follow:

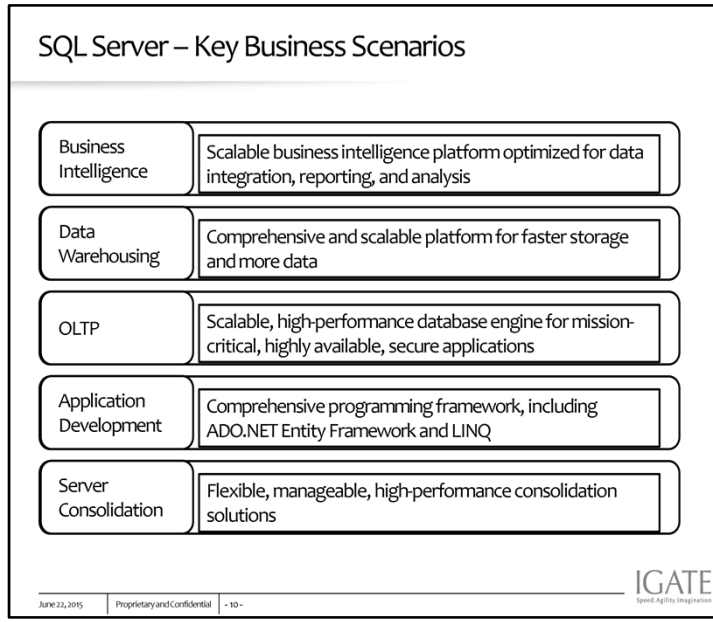
MSSQL Server Service
This Service is used to manage the database and security aspect of SQL Server.

Microsoft Search Service
This Service is for enabling the English Query Search.

SQL Server Agent
This service is for scheduling and managing tasks on SQL Server.
Microsoft Distributed Transaction Coordinator Service (MSDTCS)
This service is used for Distributed Data Management , which is an environment where data is stored at multiple locations which can be like Oracle.

SQL Server Versions

SQL Server 1.0 (1989)	Developed by Microsoft, Sybase, and Ashton-Tate for OS/2
SQL Server 4.2 (1992)	Developed for Windows NT 3.1
SQL Server 6.0 (1995)	First version designed specifically for Windows NT
SQL Server 7.0 (1999)	Total rewrite of code base resulted in performance and scalability improvements
SQL Server 2000	Further improvements in performance, scalability, and reliability
SQL Server 2005	New and improved features: Integration Services, Analysis Services, Notification Services, Reporting Services, and XML support
SQL Server 2008	Microsoft SQL Server 2008 is a powerful and reliable data management system that delivers a rich set of features, data protection, and performance for embedded application clients, light Web applications, and local data stores. Designed for easy deployment and rapid prototyping,
SQL Server 2012	SQL Server Database Engine introduces new features and enhancements that increase the power and productivity of architects, developers, and administrators who design, develop, and maintain data storage systems.



SQL Server delivers a trusted, productive, and intelligent enterprise data platform.

Business Intelligence –

SQL Server provides business intelligence functionality in various ways:

- SQL Server Database Engine provides a scalable, high-performance data-storage engine that is suitable for extremely large volumes of data.
- SQL Server Integration Services provides a comprehensive platform for extract, load, and transform (ETL) operations that enables you to incorporate disparate data into your data warehouse and business applications.
- SQL Server Analysis Services provides an online analytical processing (OLAP) engine that supports complex analytics.
- SQL Server Reporting Services makes it easy to create, publish, and distribute detailed reports.

You can easily use the 2007 Microsoft Office system to enable business users to access corporate SQL Server 2008 data. You can use the Microsoft Office Excel spreadsheet software to perform OLAP analysis, view reports, and mine data; Microsoft Office Word to view reports; and Microsoft Office SharePoint® Server 2007 to host reports.

Data Warehousing – SQL Server includes scalability features that you can use to enhance your data warehousing solutions:
• SQL Server Database Engine provides functionality to load, query, compress, and back up data.
• SQL Server Integration Services provides lookup performance and pipeline performance features that enable you to quickly load data into your data warehouse.
• SQL Server Analysis Services provides improved query processing and the ability to back up larger databases
• SQL Server Reporting Services provides improved reporting scalability.

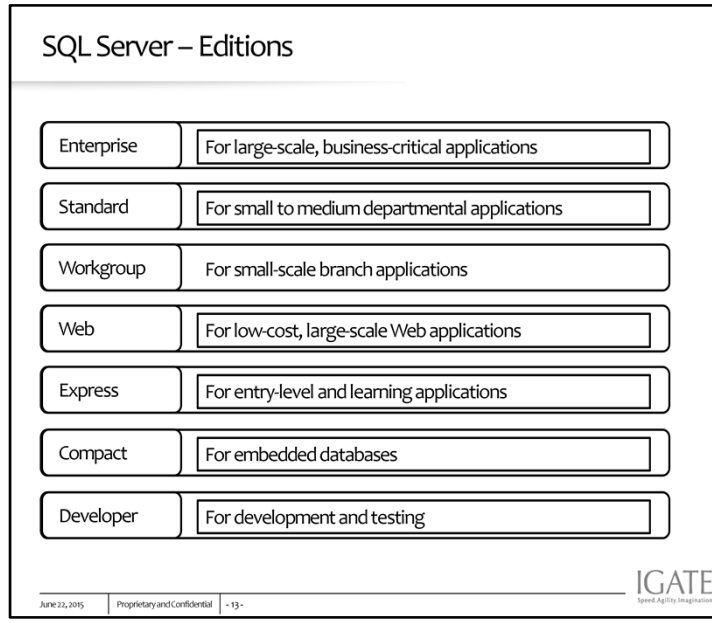
OLTP - SQL Server includes a range of features that improve the OLTP functionality of the database engine. It enables you to build scalable, high-performing, highly available, secure, and manageable database applications through the use of new and improved features such as variable-sized data types, dynamic allocation of Address Windowing Extensions (AWE), database mirroring, clustering, replication, event auditing, transparent data encryption, management policies, and centralized server administration and management.

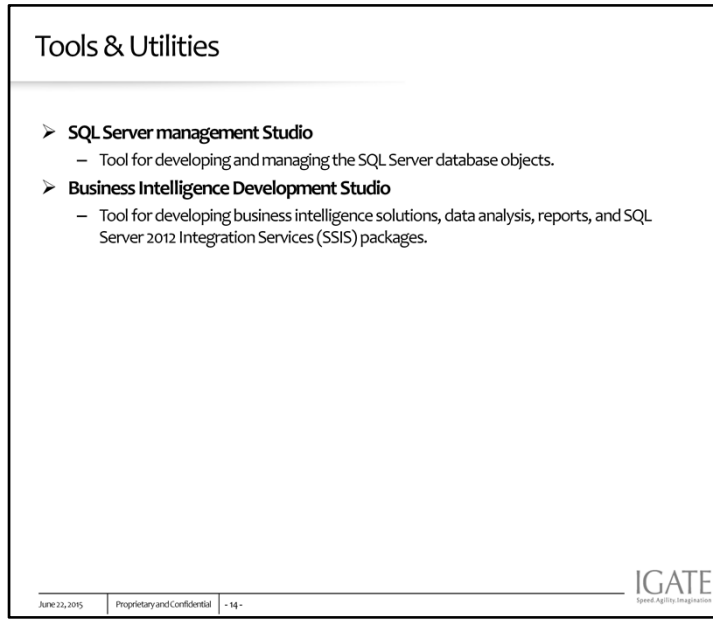
Features of SQL Server

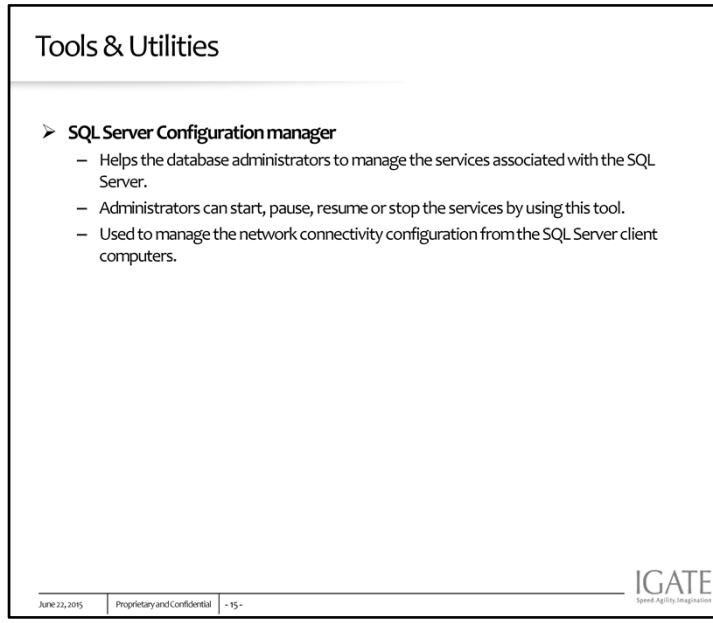
- **Scalability**
- **High level of security**
- **High Availability**
- **Manageability**
- **Programmability**
- **Replication**
- **Support for data migration and analysis**
- **Service-oriented architecture**
- **Support for Web services**
- **Reporting**
- **Data Integration**
- **Data Analysis**

Application Development - SQL Server supports the use of new programmability technologies that make developers more productive and make their applications more scalable. The introduction of the Microsoft ADO.NET Entity Framework and LINQ will simplify data access to a variety of data sources. By using the Microsoft .NET Framework common language runtime (CLR) inside the SQL Server Database Engine, developers can write easily maintained code that runs directly against the data.

Server Consolidation - SQL Server supports consolidation through several techniques, including running multiple instances of SQL Server on one physical computer and using Microsoft Virtual Server to run multiple virtual operating systems on one physical computer.


SQL Server – Editions
Enterprise
For large-scale, business-critical applications
Standard
For small to medium departmental applications
Workgroup
For small-scale branch applications
Web
For low-cost, large-scale Web applications
Express
For entry-level and learning applications
Compact
For embedded databases
Developer
For development and testing
June 22, 2015
Proprietary and Confidential
- 13 -
IGATE
Speed. Agility. Imagination


Tools & Utilities
SQL Server management Studio
Tool for developing and managing the SQL Server database objects.
Business Intelligence Development Studio
Tool for developing business intelligence solutions, data analysis, reports, and SQL Server 2012 Integration Services (SSIS) packages.
June 22, 2015
Proprietary and Confidential
- 14 -
IGATE
Speed. Agility. Imagination

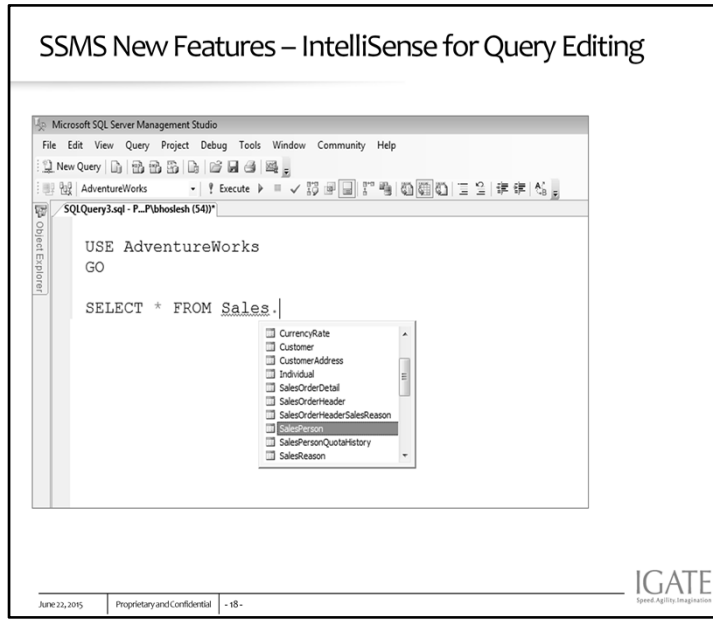

Tools & Utilities
SQL Server Configuration manager
– Helps the database administrators to manage the services associated with the SQL Server.
– Administrators can start, pause, resume or stop the services by using this tool.
– Used to manage the network connectivity configuration from the SQL Server client computers.
June 22, 2015
Proprietary and Confidential
- 15 -
IGATE
Speed. Agility. Imagination

Tools & Utilities

- **Database Engine Tuning Advisor**
  - GUI tool used to provide tuning recommendations.
  - Acts as a tuning advisory.
  - Tuning information is stored in the msdb database
- **SQL Server Profiler**
  - Graphical user interface to SQL Trace for monitoring an instance of the SQL Server Database Engine or Analysis Services.
  - Used to capture and save data about each event to a file or table to analyze later.

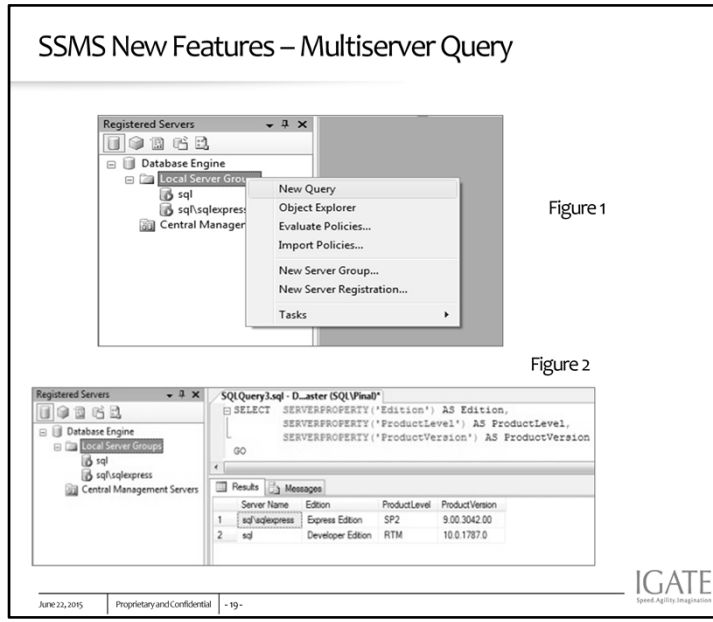
SQL Server – SSMS New Features

- **The SQL Server 2008 Management Studio (SSMS) has several new features which can help you administer your server, write queries or develop applications**
- **You can use these Features to work more efficiently**
  - IntelliSense for Query Editing
  - Multi Server Query
  - Query Editor Regions
  - Object Explorer Enhancements



IntelliSense

The Database Engine Editor now provides IntelliSense functionality such as word completion, error underlining, parameter help, colorizing, Quick Info, outlining, and syntax pair matching. IntelliSense is provided for frequently used Transact-SQL elements. It will be extended to other Transact-SQL elements in future releases.



MultiServer Query

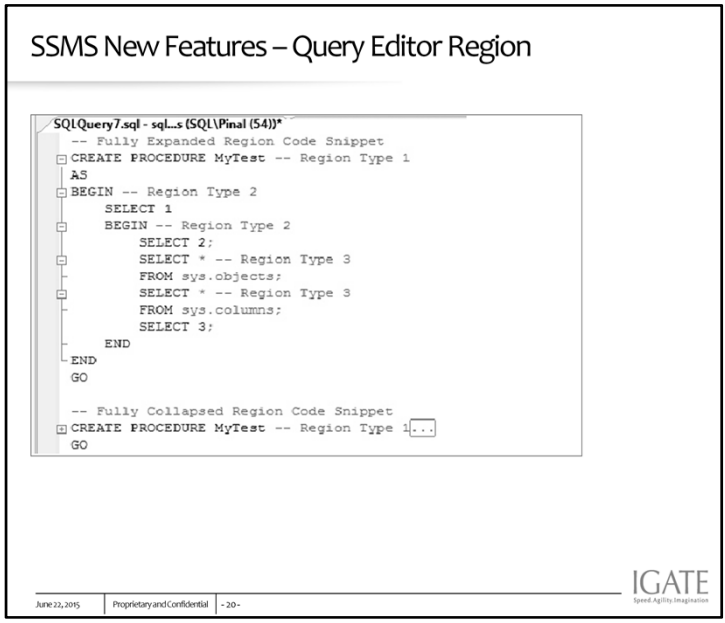
Usually DBA don't manage only one database; they have many servers to manage. There are cases when DBA has to check the status of all the servers.

SSMS 2008 has a feature to run a query on different servers from one query editor window. First of all make sure that you registered all the servers under your registered server. Once they are registered Right Click on server group name and click New Query as shown in the image Figure1 above.

Now in the opened query window run the following query

```
SELECT
SERVERPROPERTY('Edition') AS Edition,
SERVERPROPERTY('ProductLevel') AS ProductLevel,
SERVERPROPERTY('ProductVersion') AS ProductVersion
```

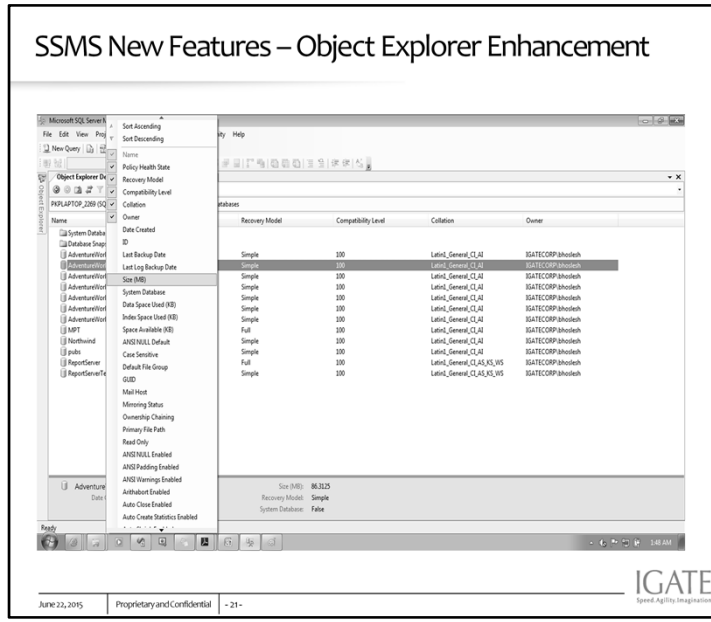
Query above will give the result shown in the image above. Note that we have only three columns in the SELECT but our output contains four columns. The very first column is the "Server Name" and it is added by SQL Server to indentify rows belonging to a specific server.



Query Editor Regions

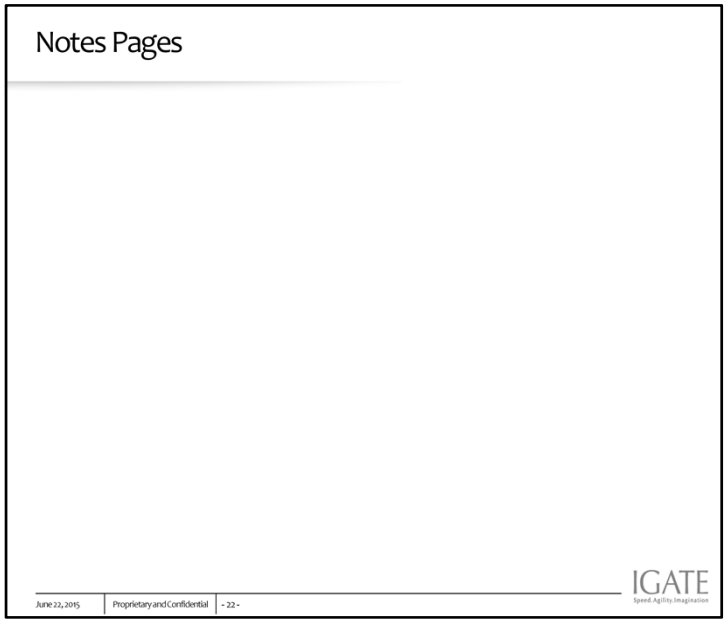
This feature exists in many programming languages already but now it is newly introduced in SSMS 2008. There are cases when T-SQL code is longer than hundreds of lines and after a while it keeps on getting confusing.

Regions can be collapsed as well as expanded by clicking the small sign of '-' or '+' besides them. The following image shows a fully expanded region code snippet and a fully collapsed code snippet.



Object Explorer Enhancements

Object Explorer Detail view can be enabled by either going to Menu >> View >> Object Explorer Detail or pressing F7. In Object Explorer Detail the new feature is Object Search. Enter any object name in the object search box and the searched result will be displayed in the same window as Object Explorer Detail.



SQL Server 2012 – SSMS New Features

SQL Server 2012 brings some of the most desired enhancements in the Management Studio to simplify manageability.

The new SQL Server Management Studio is built on Visual Studio 2012. As it opens, you soon recognize the familiar dark blue theme of Visual Studio.

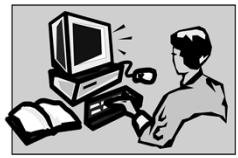
You can drag and move out Query Window from the main work area.

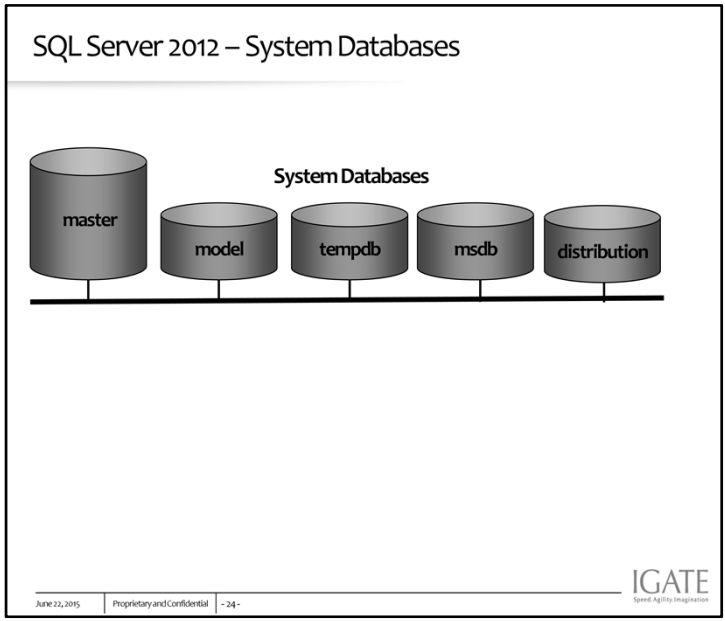
You can even drag out the ToolBox from the main IDE and place it elsewhere on the desktop or on another monitor

Another important feature is *Insert Snippet* which you can find on menu via Edit -> IntelliSense. It allows you to inject variety of code snippets onto your Query Window:

Demo

- **SSMS New Features**





Types of Databases

SQL Server databases are categorized as below :

System databases.

User-defined databases.

Figure above displays the two categories of databases.

System Databases

A system database is created to support the operation of the SQL server. When SQL Server is installed , four system databases – Master, Model, Tempdb, Msdb and two user databases are created automatically.

These databases are discussed as :

Master database

The master database is the basic database used by SQL Server in its operation. The master database controls the user databases and the operation of SQL Server as a whole. The default size of this database is 12.25 MB. This database contains pointers to all other databases stored on your system. It also includes server-wide information such as login information, system stored procedures and related services. Since the information stored in the master database is very critical in nature, no user is allowed direct access to the master database, which is updated on a continuous basis.

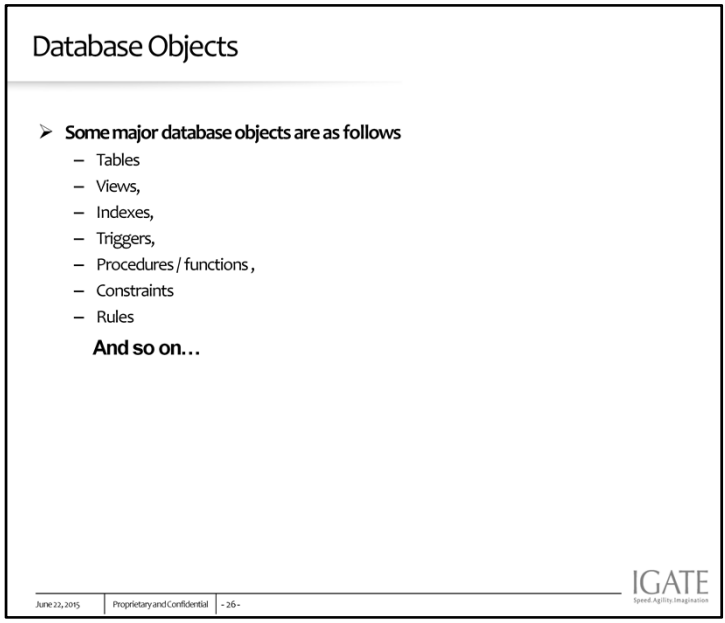
Model database
The model database is the template database for user – defined databases in SQL Server. This database consists of the system tables, which should belong to every user databases. The structure of the model database is copied into each new database that is created by the database administrator (DBA). Hence ,whenever a new database is created its size should be at least the same as size of model database, so that the structure of the model database can be accustomed in new database.

Mssdb database
The msdb database contains task scheduling, event handling, replication, alert management and system operator information (all these keywords would be discussed in detail in the later chapters).

Tempdb database
The tempdb database provides storage area from temporary tables, as also the result of sort operations, join operations, and other activities that require temporary space to execute. There is only one tempdb database regardless of the number of the databases stored on the server. No special permission is required to use this database. They are deleted when the user logs out of the SQL Server session.
There are two types of temporary tables that are supported by SQL Server as mention below:
Global temporary tables: These tables are available to all the users that are connected and are prefixed with double–hash signs(##), to the name of the table.
Local temporary tables: These tables are only visible to the user who have created these and are prefaced with a single hash (#) sign.
The information about the various system databases can be summarized in Table below
Table: System Databases

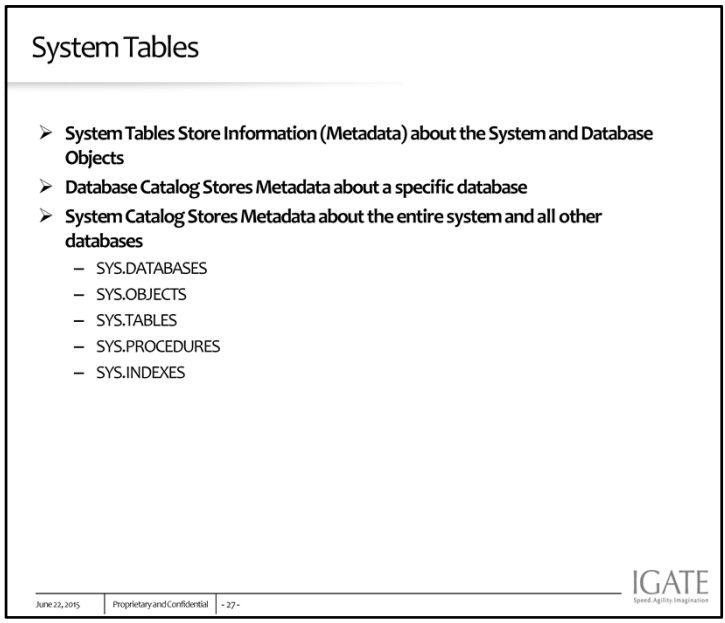
Databases	**Class**	**Size (in MB)**
Master	System databases	12.25
Model	System databases	1.5
Tempdb	System databases	8.5
Msdb	System databases	

User Databases
A user database is a database that is created by the Database Administrator or a user. User defined databases are those created as per the needs of the users of SQL Server.
An example of the sample user database is the **AdveentureWorks** database.



A database can also be defined as a collection of interrelated data. The data is stored in the form of different objects that allow the users to store and retrieve information. These on objects help in structuring different objects that allow the users to store and retrieve information. These objects help in structuring data and in defining data integrity mechanisms. The different database objects are:

Database Object	Description
Table	It is a fundamental method of storing data in the form of a collection of columns and rows (records).
Data Types	An identifier that specifies the nature of data that can be stored in a column.
Defaults	Specifies a value to be assigned by SQL Server for a column if the user does not enter any value
Index	A structure that helps faster access to records of a table based on the values of one or more columns. It contains data values and pointers to the records where those values occur.
Constraint	Is a restriction that is enforced by SQL Server to limit the values that the user can enter into a column.
Rule	Specifies acceptable values that can be inserted into a particular column.
Stored Proc	A pre-complied collection of SQL statements, which are used to perform specific tasks.
Trigger	A special type of stored procedure that is automatically executed by SQL Server in response to an DML statement.
View	A virtual table that provides an alternate way to look at data from one or more tables in a database.



System Catalog:
The system tables that SQL server needs to configure and manage itself are collectively grouped together and called the "System Catalog". The **system catalog** consists primary of the system tables contained in the master table.

Database Catalog
The system tables that are needed to define the structure of a new user database are collectively known as the "Database Catalog". The database catalog consists primarily system tables contained in the model database.
The various system tables in all the SQL Server system databases are discussed below as follows:

Some MASTER system Tables
Syslockinfo - Contains information on all granted, converting, and waiting lock requests.
Syscacheobjects - Contains information about how the cache is used.
Syslogins - Contains one row for each login account.
Sysdatabases - Contains one row for each database on SQL server 2008.

Model System Tables

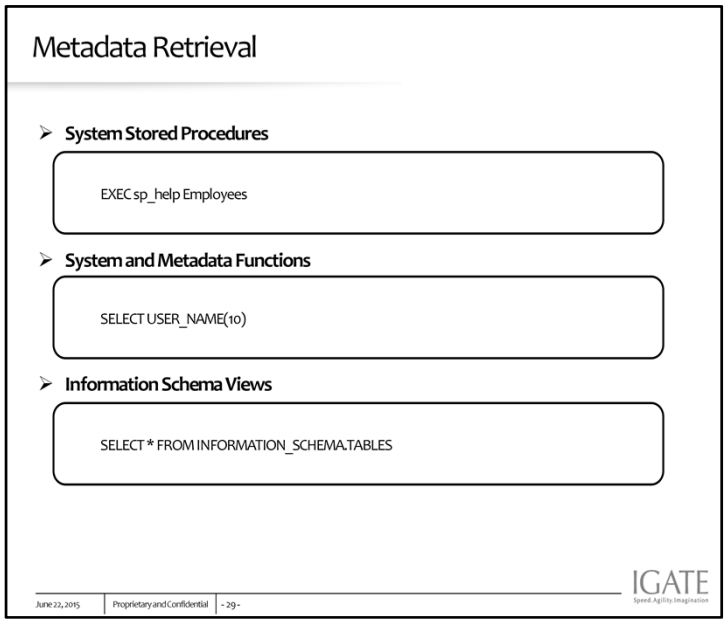
Name	Utility
Syscolumns	Contains one row for every column in every table and view, and a row for each parameter in a stored procedure.
Syscomments	Contains entries for each view, rule, default, trigger, CHECK constraint , DEFAULT constraint , and stored procedure.
Sysfiles	Contains one row for each file in a database
Sysindexes	Contains one row for each index and table in the database
Sysobjects	Contains one rows for each object (Constraint , default, log, rule, stored procedure, and so on) created within a database. Syscomments and Sysobjects, combined together, give detailed information on ny object in SQL Server.
Syspermissions	Contains information about permissions granted and denied to users , groups and roles in the database.
Systypes	Contains one row for each system-supplied and each user-defined data type
Sysusers	Contains one row for each windows user, windows group, SQL server user, or SQL Server role in the database

MSDB System Tables

Name	Utility
Sysalerts	Contains one row for each alert. An alert is a message sent in response to an event.
Syscategories	Contains the categories used by SQL Server Enterprise Manager to organize jobs, alerts, and operators.
Sysobs	Stores the information for each scheduled job to be executed by the SQL Server Agent
Sysoperators	Contains one row for each SQL Server operator.
Sysnotification	Contains one row for each notification to an operator

TEMPDB System Tables

Name	Utility
Local Temporary tables	These tables are only visible to the user who have created these and are prefaced with a single hash (#) sign.
Global Temporary tables	These tables are available to all the users that are connected and are prefaced with two-hash signs(##).



System Stored Procedures

Many of your administrative activities in SQL Server are performed through a special kind of procedure known as a **system stored procedure**. System stored procedures are created and stored in the **master** database and have the **sp_** prefix. System stored procedures can be executed from any database without having to qualify the stored procedure name fully using the database name **master**.

It is strongly recommended that you do not create any stored procedures using **sp_** as a prefix. SQL Server always looks for a stored procedure beginning with **sp_** in this order:

The stored procedure in the **master** database.

The stored procedure based on any qualifiers provided (database name or owner).

The stored procedure using **dbo** as the owner, if one is not specified.

Therefore, although the user-created stored procedure prefixed with **sp_** may exist in the current database, the **master** database is always checked first, even if the stored procedure is qualified with the database name.

Important

If any user-created stored procedure has the same name as a system stored procedure, the user-created stored procedure will never be executed.

System Procedures

sp_add_data_file_recover_suspect_db, sp_add_log_file_recover_suspect_db sp_helpconstraint, sp_addextendedproc, sp_helpdb, sp_addextendedproperty, sp_helpdevice, sp_helpextendedproc, sp_addmessage, sp_helpfile,sp_addtype

Summary

- **In this lesson, you have learnt:**
  - SQL Server 2012 database follows the relational model of data management system
  - SQL Server databases are categorized as:
    - System databases.
    - User-defined databases
  - SQL Server offers various Tools and Utilities to work with

Review Question

- **Question 1: List the two authentication modes available in MS SQL Server.**
- **Question 2: List the System Databases in SQL Server.**