SQL Server 2012 – Database Development

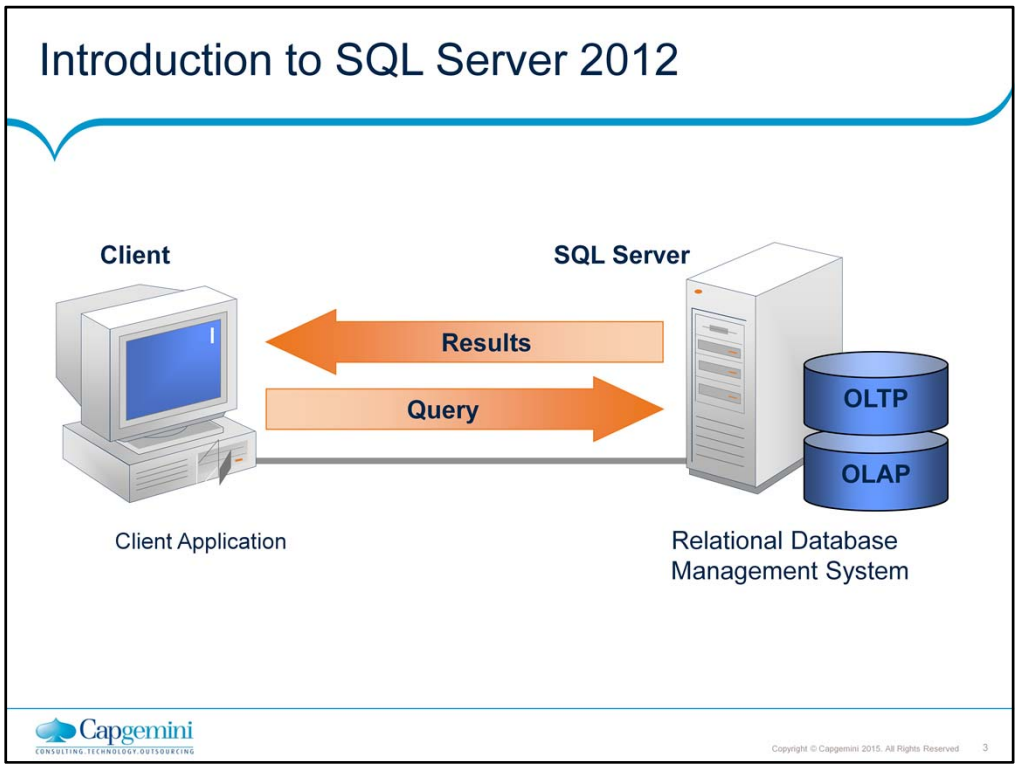
Lesson 2: Introduction to SQL Server 2012

Lesson Objectives

- In this lesson, you will learn:
  - What Is SQL Server?
  - SQL Server Components
  - SQL Server Versions, Editions and Tools
  - SQL Server Databases
  - Features of SQL Server

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 2



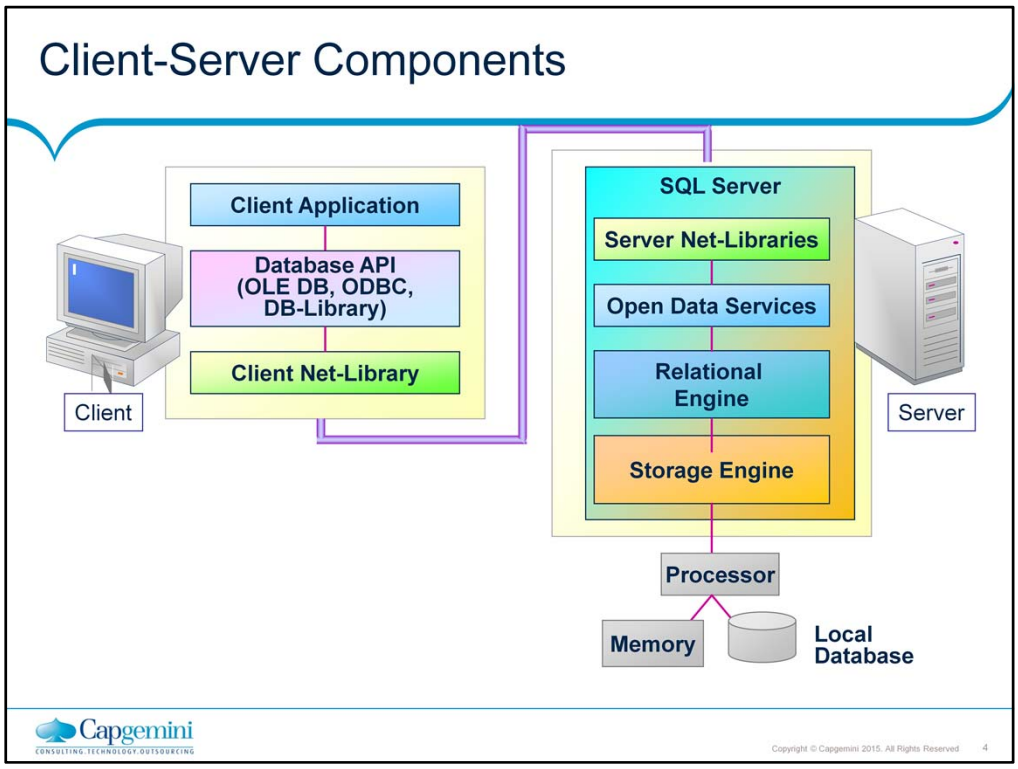
What is SQL Server 2012?

SQL Server 2012 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 2012 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.

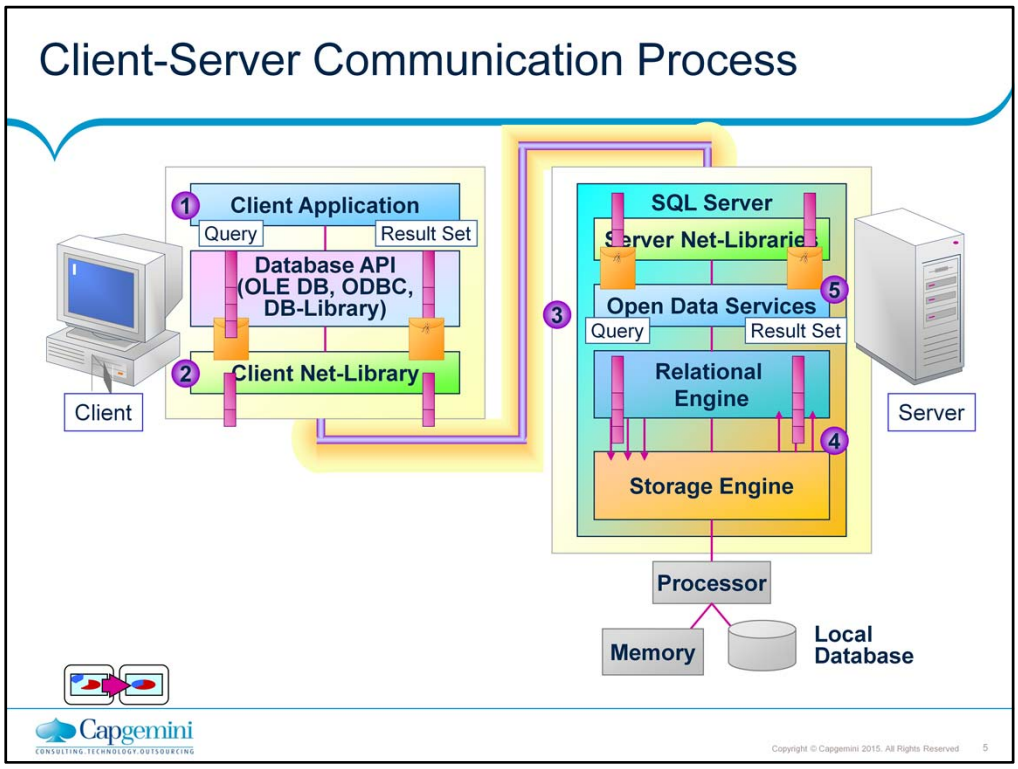


Figure shows the path from the SQL Server client application to the SQL Server engine and shows where the Net-Library interface fits in. On the server side, ODS provides functionality that mirrors that of ODBC, OLE DB, or DB-Library at the client. Calls exist for an ODS server application to describe 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:

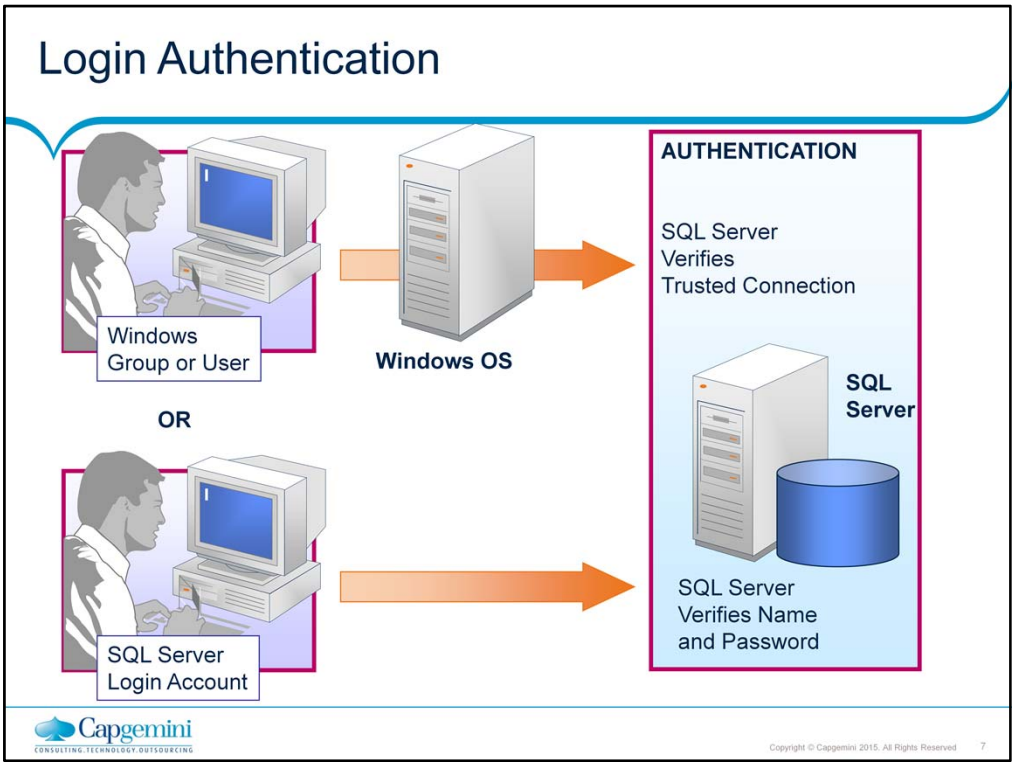
```
SELECT CompanyName, OrderID, ShippedDate
FROM Northwind.dbo.Customers AS Cst
JOIN Northwind.dbo.Orders AS Ord
ON (Cst.CustomerID = Ord.CustomerID)
```

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.

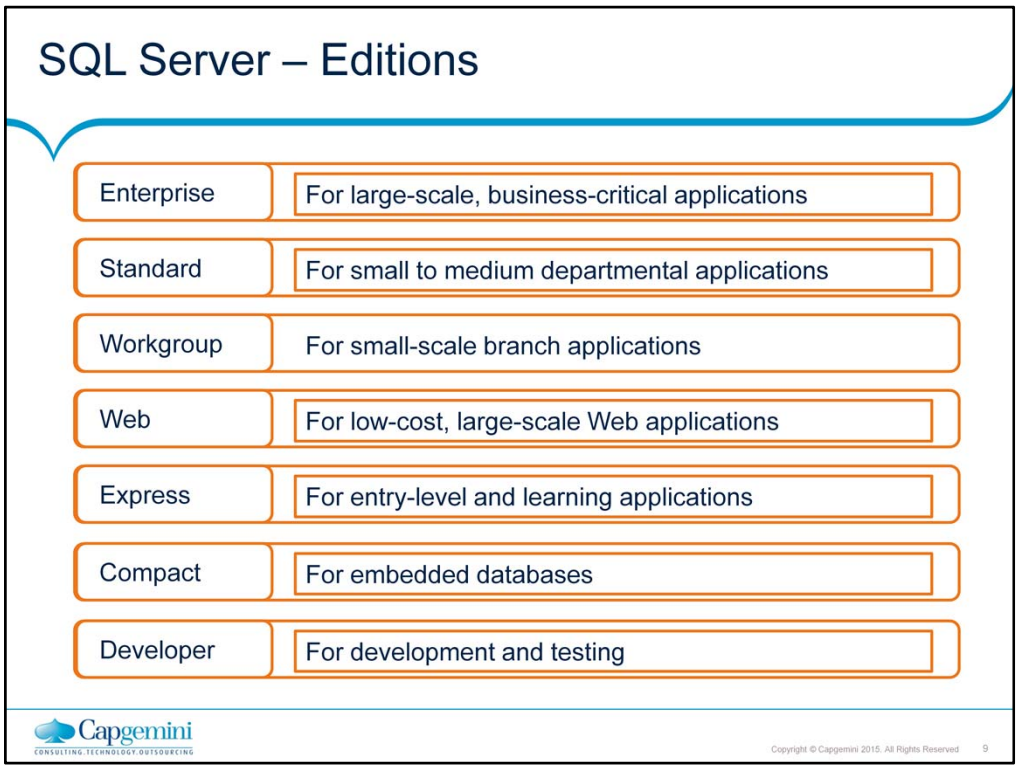
Mixed Mode (Windows Authentication and SQL Server Authentication) Mixed Mode allows users to connect to an instance of SQL Server using either Windows Authentication or SQL Server Authentication. Users who connect through a Windows user account can make use of trusted connections in either Windows Authentication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

SQL Server 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.

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 8


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
Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING
Copyright © Capgemini 2015. All Rights Reserved
9

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.

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 10

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.

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

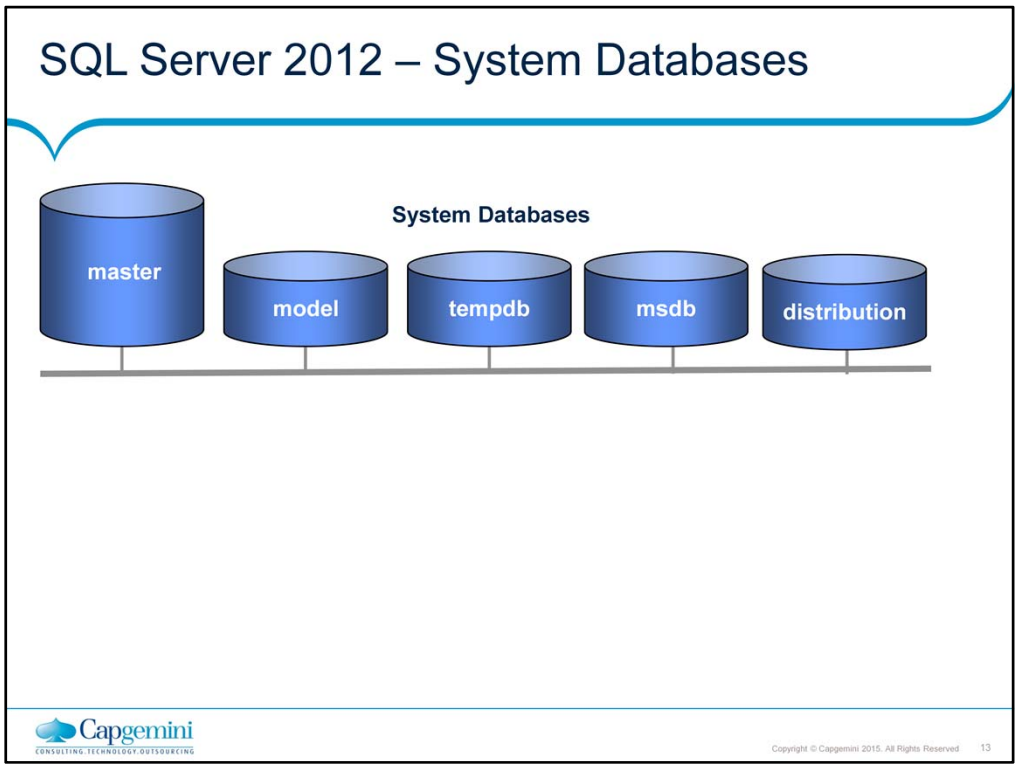
 11

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.

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 12



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.

Msdb 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

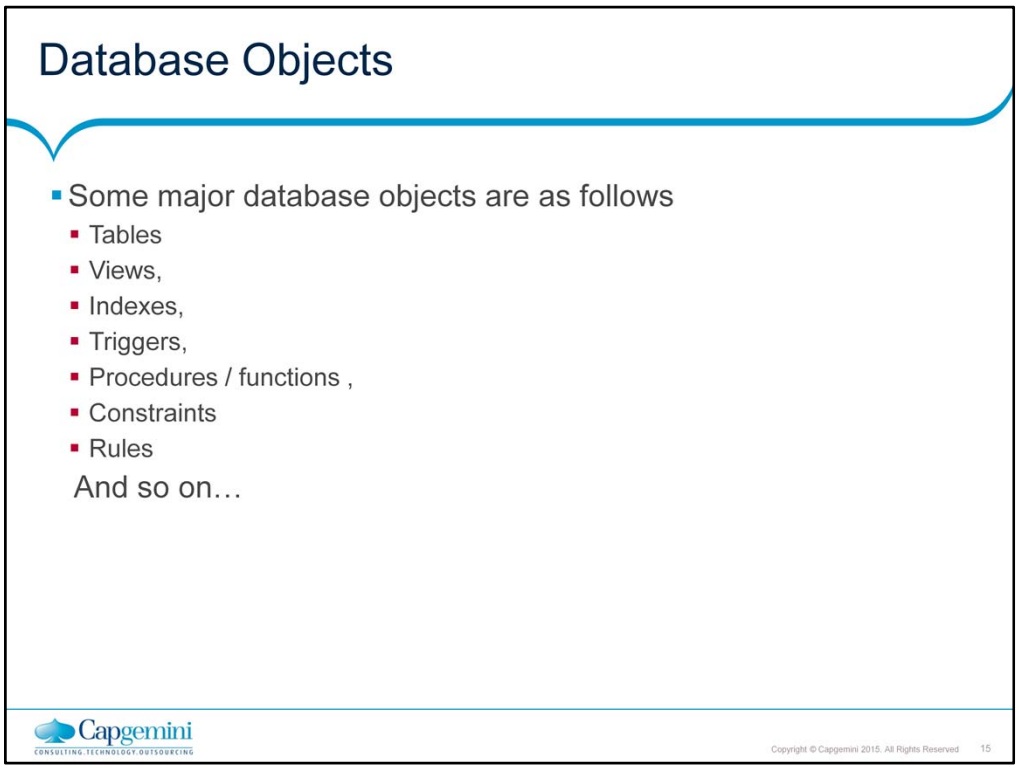
Table: System Databases

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

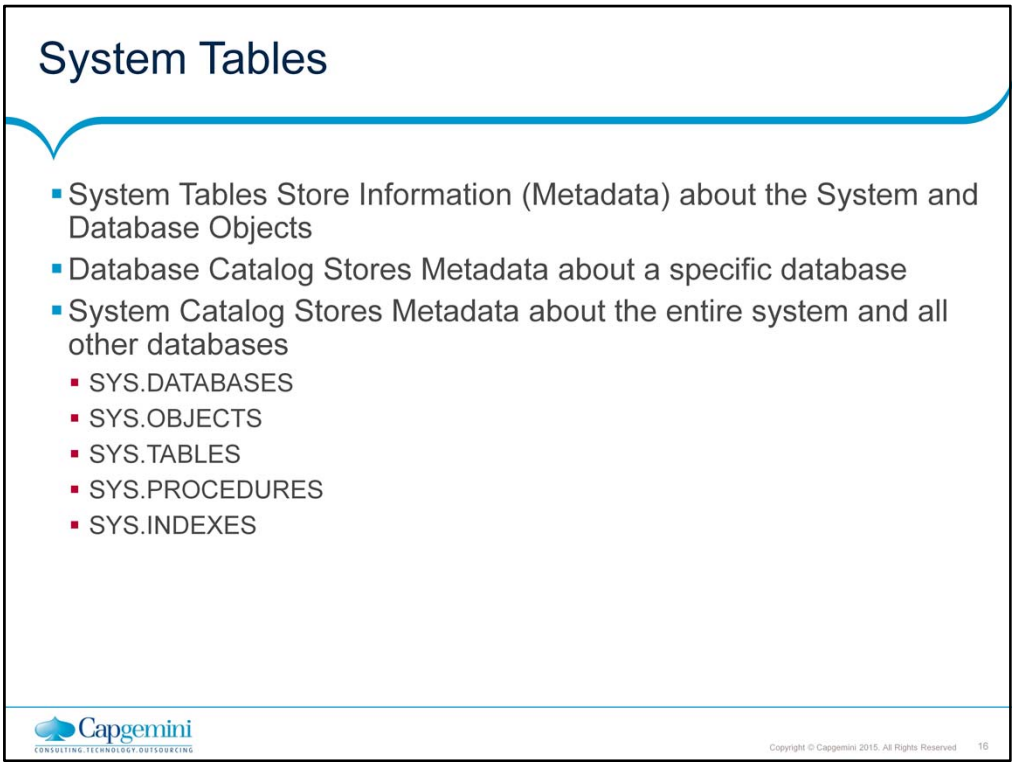
A user database is a database that is created by the Database Administrator or a user. User defined databases are those created as per the needs of the users of SQL Server.

An example of the sample user database is the **AdventureWorks** 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.

```
SELECT * FROM Syslockinfo
```

Syscacheobjects - Contains information about how the cache is used.

```
SELECT * FROM Syscacheobjects
```

Syslogins - Contains one row for each login account.

```
SELECT * FROM Syslogins
```

Sysdatabases - Contains one row for each database on SQL server 2012.

```
SELECT * FROM Sysdatabases
```

Model System Tables

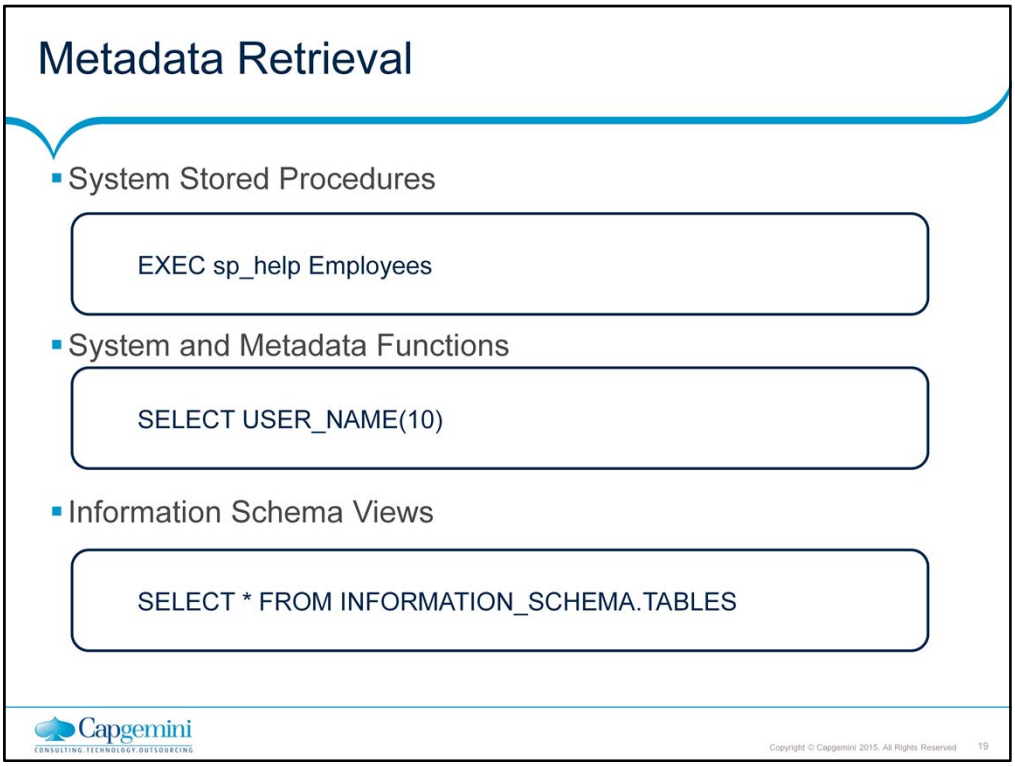
Name	Utility
Syscolumns	Contains one row for every column in every table and view, and a row for each parameter in a stored procedure. SELECT * FROM Syscolumns
Syscomments	Contains entries for each view, rule, default, trigger, CHECK constraint , DEFAULT constraint , and stored procedure. SELECT * FROM Syscomments
Sysfiles	Contains one row for each file in a database SELECT * FROM Sysfiles
Sysindexes	Contains one row for each index and table in the database SELECT * FROM Sysindexes
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 my object in SQL Server. SELECT * FROM Sysobjects
Syspermissions	Contains information about permissions granted and denied to users , groups and roles in the database. SELECT * FROM Syspermissions
Systypes	Contains one row for each system-supplied and each user-defined data type SELECT * FROM Systypes
Sysusers	Contains one row for each windows user, windows group, SQL server user, or SQL Server role in the database SELECT * FROM Sysusers

MSDB System Tables

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

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

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 20

Review Question

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

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 21