CS108: Advanced Database

- Database Programming

Lecture 01:

Introduction

Course Overview

- Introduces relational and object databases to support database creation and application development
  - T-SQL, database design and application development
- Techniques for web based data retrieval and manipulation
- Requires practice
  - Introduction to Database Systems
  - Data Structures

Syllabus

- Database Programming
  - Transact-SQL
  - Controlling Execution
  - T-SQL Cursors
  - Stored Procedures
  - Triggers
  - Etc.
- ASP or ASP.NET
  - Web Form
  - Controls
  - Etc.

Course Information

- *Lecture Hours and Venues*
  - TUE 11:00 – 12:45 N416
  - FRI 15:00 – 16:45 C408
- *Course FTP*
  - ftp://ftp.must.edu.mo
  - name: sllo_stu
  - password: Q0xTDh6Q

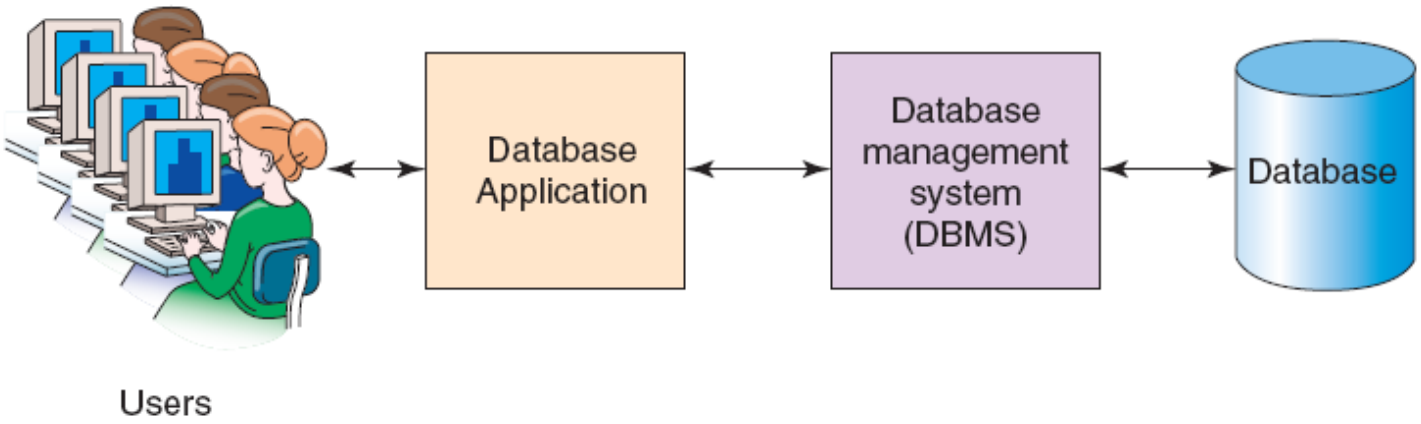
Course Assessment

- Class Participation ~10%
- Assignments and Labs ~10%
- Quiz ~30%
- Project/Final Report ~50%

RDBMS BASIC
SQL Server Database

Relational Database Management System

- Advanced RDBMSs
  - Store the data
  - Manage the data
  - Restricting the kind of data that can go into the system
  - Facilitating getting data out of the system



Where are RDBMS used ?

- Backend for traditional “database” applications
- Backend for large Websites
- Backend for Web services

Functionality of a DBMS

The programmer sees SQL, which has two components:

- Data Definition Language - DDL
- Data Manipulation Language - DML

Behind the scenes the DBMS has:

- Query engine
- Query optimizer
- Storage management
- Transaction Management (concurrency, recovery)

How the Programmer Sees the DBMS

- Start with DDL to *create tables*:

```
CREATE TABLE Students (
        Name CHAR(30)
        SSN CHAR(9) PRIMARY KEY NOT NULL,
        Category CHAR(20)
)   . . .
```

- Continue with DML to *populate tables:*

```
INSERT INTO Students
VALUES('Charles', '123456789', 'undergraduate')
. . . .
```

How the Programmer Sees the DBMS

- Tables:

Students:

SSN	Name	Category
123-45-6789	Charles	undergrad
234-56-7890	Dan	grad
	…	…

Takes:

SSN	CID
123-45-6789	CSE444
123-45-6789	CSE444
234-56-7890	CSE142
	…

Courses:

CID	Name	Quarter
CSE444	Databases	fall
CSE541	Operating systems	winter

- Still implemented as files, but behind the scenes can be quite complex

"*data independence*" = separate *logical* view from *physical implementation*

Queries

- Find all courses that “Mary” takes

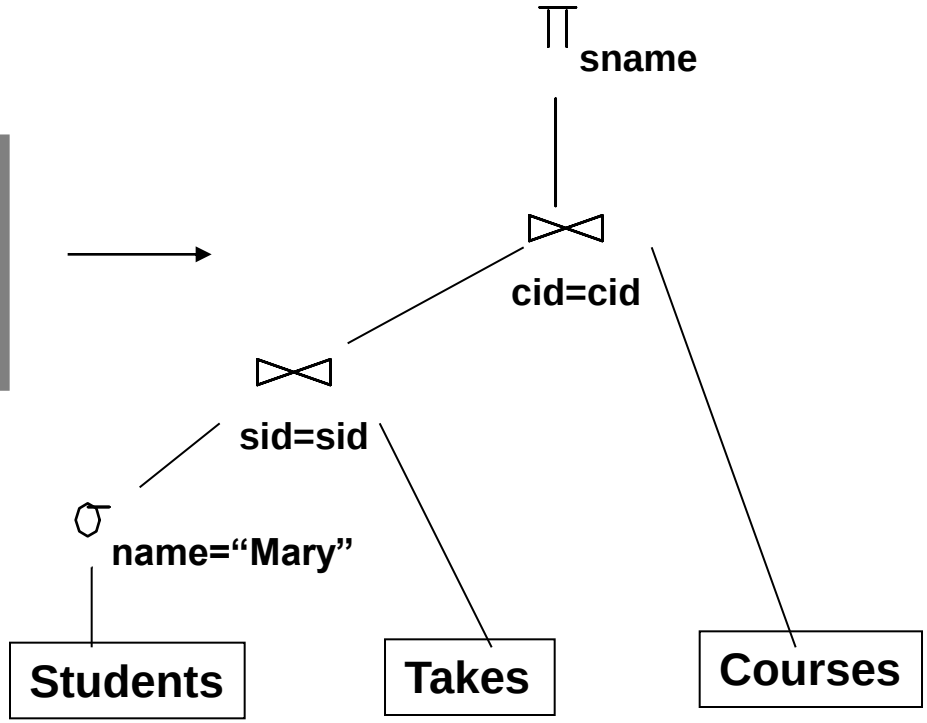
```
SELECT  C.name
FROM    Students S, Takes T, Courses C
WHERE   S.name=“Mary” and
        S.ssn = T.ssn and T.cid = C.cid
```

- What happens behind the scene ?
  - Query processor figures out how to answer the query efficiently.

Queries, behind the scene

Declarative SQL query → *Imperative query execution plan:*

```
SELECT  C.name
FROM Students S, Takes T, Courses C
WHERE S.name="Mary" and
        S.ssn = T.ssn and T.cid = C.cid
```



The **optimizer** chooses the best execution plan for a query

Transactions

- A *transaction* = sequence of statements that either all succeed, or all fail
- Transactions have the ACID properties:

  A = atomicity

  C = consistency

  I = isolation

  D = durability

Transactions

- Enroll “Mary Johnson” in “CS108”:

```
BEGIN TRANSACTION;

INSERT INTO Takes
    SELECT Students.SSN, Courses.CID
    FROM Students, Courses
    WHERE Students.name = ‘Mary Johnson’ and
                Courses.name = ‘CS108’

-- More updates here....

IF everything-went-OK
     THEN COMMIT;
ELSE ROLLBACK
```

- If system crashes, the transaction is still either committed or aborted

SQL Server Overview

- SQL Server is a relational database management and analysis system by Microsoft
  - Targeting the enterprise-level database market
- Major versions
  - SQL Server 7.0 (Windows NT)
  - SQL Server 2000 (Windows Server 2000)
  - SQL Server 2005 (Windows Server 2003)
  - SQL Server 2008 (Windows Server 2008)
  - SQL Server 2008 R2 (Windows Server 2008)
  - SQL Server 2012

The Database Object

- The database is effectively the highest-level object that you can refer to within a given SQL Server
- Database Objects
  - The database itself
  - The transaction log
  - Tables, Indexes, Views
  - Stored procedures
  - Sequences
  - User-defined functions
  - User-defined data types

The Database Object

- A database is typically a group of constructs that include
  - A set of table objects
  - Other objects, such as stored procedures
  - Views that pertain to the particular grouping of data stored in the database's tables
- When we first load SQL Server, we start with at least four system databases:
  - master
  - model
  - msdb
  - tempdb

The master Database

- Every SQL Server, regardless of version or custom modifications, has the master database
- Holds a special set of tables (system tables) that keeps track of the system as a whole
  - for example, when we create a new database on the server, an entry is placed in the **sysdatabases** table in the master database
- All extended and system-stored procedures, regardless of which database they are intended for use with, are stored in this database.
- The master database **cannot** be deleted.

The model Database

- The model database is aptly named, in the sense that it's the model on which a copy can be based.
- The model database forms a template for any new database that we create.
- If we want, alter the model database if we want to change what standard, newly created databases look like.
  - for example, we could add a set of audit tables that we include in every database we build.
- Note that because this database serves as the template for any other database, it's a required database and must be left on the system; we **cannot** delete it.

The msdb Database

- msdb is where the SQL Agent process stores any system tasks.
- If we schedule backups to run on a database nightly, there is an entry in msdb. Schedule a stored procedure for one-time execution, and yes, it has an entry in msdb.
- Other major subsystems in SQL Server make similar use of msdb.

The tempdb Database

- tempdb is one of the key working areas for our server.
  - A complex or large query that SQL Server needs to build interim tables to solve, it does so in tempdb
  - Create a temporary table of our own, it is created in tempdb
  - A need for data to be stored temporarily, it’s probably stored in tempdb
- tempdb is very different from any other database. The objects within it temporary, the database itself is also temporary.
- tempdb has the distinction of being the only database in our system that is rebuilt from scratch every time we start our SQL Server

The Transaction Log

- Any changes we make don’t initially go to the database itself. Instead, they are written serially to the transaction log.
- At some later point in time, the database is issued a checkpoint; it is at that point in time that all the changes in the log are propagated to the actual database file.
- The database is in a random access arrangement, but the log is serial in nature.
- The log accumulates changes that are deemed as having been committed, and then writes several of them at a time to the physical database file(s).

Table

- A table consists of columns and rows. The actual data for the database is stored in the tables.
- Each table definition also contains the metadata (descriptive information about data) that describes the nature of the data.
- Each column has its own set of rules about what can be stored in that column.
- A violation of the rules of any one column can cause the system to reject an inserted row, an update to an existing row, or the deletion of a row.

Indexes

- An index is an object that exists only within the framework of a particular table or view
- An index works much like the index does in the back of an encyclopedia
- An index provides we ways of speeding the lookup of our information
- Indexes fall into two categories:
  - Clustered
  - Non-clustered

Triggers

- A trigger is an object that exists only within the framework of a table
- Triggers are pieces of logical code that are automatically executed when certain things (such as inserts, updates, or deletes) happen to our table
- Triggers can be used for a great variety of things, but are mainly used for either copying data as it is entered or checking the update to make sure that it meets some criteria

Constraints

- A constraint is yet another object that exists only within the confines of a table
- Constraints are much like they sound; they confine the data in our table to meet certain conditions
- Constraints, in a way, compete with triggers as possible solutions to data integrity issues
- They are not, however, the same thing: Each has its own distinct advantages

Views

- A view is something of a virtual table
- A view, for the most part, is used just like a table, except that it doesn't contain any data of its own
- Instead, a view is merely a preplanned mapping and representation of the data stored in tables
- A view is stored in the database in the form of a *query*
  - This query calls for data from some, but not necessarily all, columns to be retrieved from one or more tables

Stored Procedures

- Stored procedures are the bread and butter of programmatic functionality in SQL Server
- Stored procedures are generally an ordered series of Transact-SQL statements bundled up into a single logical unit
  - Allow for variables and parameters
  - Selection and looping constructs

User-Defined Functions

- User-defined functions (UDFs) have a tremendous number of similarities to stored procedures, except:
  - Can return a value of most SQL Server data types
  - Can’t have side effects. For example, can’t changing tables
- UDFs are similar to the functions that we would use in a standard programming language such as C++
  - We can pass more than one variable in and get a value out

SQL Server Data Types

- SQL Server has for one of the fundamental items of any environment that handles data – data types
- Most of these have equivalent data types in other programming languages
  - for example, an int in SQL Server is equivalent to a signed int in C++ for most systems and compiler combinations

SQL Server Data Types

- SQL Server data types work much as you would expect given experience in most other modern programming languages
  - Adding numbers yields a sum, but adding strings concatenates them
  - When we mix the usage or assignment of variables or fields of different data types, a number of types convert implicitly (or automatically)

SQL Server Data Types

Data Type	Class	Size	Nature of the Data
Bit	Integer	1	Boolean, two states, for example, T/F
Bigint	Integer	8	–263 to 263–1
Int	Integer	4	–2,147,483,648 to 2,147,483,647
SmallInt	Integer	2	–32,768 to 32,767
TinyInt	Integer	1	0 to 255
Decimal or Numeric	Decimal or Numeric	Varies	Fixed precision and scale from $-10^{38}-1$ to $10^{38}-1$
Money	Money	8	Monetary units from –263 to 263 plus precision to four decimal places.
SmallMoney	Money	4	–214,748.3648 to +214,748.3647
Float	Approximate Numerics	Varies	Ranges from –1.79E + 308 to 1.79E + 308.
DateTime	Date/Time	8	Date and time data from January 1, 1753, to December 31, 9999, with an accuracy of three hundredths of a second.

Data Type	Class	Size	Nature of the Data
DateTime2	Date/Time	Varies (6–8)	Updated incarnation of the more venerable DateTime data type.
SmallDateTime	Date/Time	4	Date and time data from January 1, 1900, to June 6, 2079, with an accuracy of one minute.
DateTimeOffset	Date/Time	Varies (8–10)	Similar to the DateTime data type.
Date	Date/Time	3	Stores only date data from January 1, 0001, to December 31, 9999, as defined by the Gregorian calendar.
Time	Date/Time	Varies (3–5)	Stores only time data in user-selectable precisions as granular as 100 nanoseconds
Cursor	Special Numeric	1	Pointer to a cursor.
Char	Character	Varies	Fixed-length character data. Values shorter than the set length are padded with spaces to the set length. Data is non-Unicode. Maximum specified length is 8,000 characters.

Data Type	Class	Size	Nature of the Data
VarChar	Character	Varies	Variable-length character data. Values are not padded with spaces.
NChar	Unicode	Varies	Fixed-length Unicode character data.
NVarChar	Unicode	Varies	Variable-length Unicode character data.
Binary	Binary	Varies	Fixed-length binary data with a maximum length of 8,000 bytes.
VarBinary	Binary	Varies	Variable-length binary data with a maximum specified length of 8,000 bytes.
Table	Other	Special	This is primarily for use in working with result sets, typically passing one out of a User-Defined Function or as a parameter for stored procedures. Not usable as a data type within a table definition.
XML	Character	Varies	Defines a character field as being for XML data. Provides for the validation of data against an XML Schema, as well as the use of special XML-oriented functions.

Type Conversions

	binary	varbinary	char	varchar	nchar	nvarchar	datetime	smalldatetime	date	time	datetimeoffset	datetime2	decimal	numeric	float	real	bigint	int(INT4)	smallint(INT2)	tinyint(INT1)	money	smallmoney	bit	timestamp	uniqueidentifier	image	ntext	text	sql_variant	xml	CLR UDT	hierarchyid
binary		●	●	●	●	●	●	●	●	●	●	●	●	●	○	○	●	●	●	●	●	●	●	●	●	●	○	○	●	●	●	●
varbinary	●		●	●	●	●	●	●	●	●	●	●	●	●	○	○	●	●	●	●	●	●	●	●	●	●	○	○	●	●	●	●
char	●	●		●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
varchar	●	●	●		●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
nchar	●	●	●	●		●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
nvarchar	●	●	●	●	●		●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	○	●	●	●	●	●	●
datetime	●	●	●	●	●	●		●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	○	○	○	○	●	○	○	○
smalldatetime	●	●	●	●	●	●	●		●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	○	○	○	○	●	○	○	○
date	●	●	●	●	●	●	●	●		○	●	●	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	●	○	○	○
time	●	●	●	●	●	●	●	●	○		●	●	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	●	○	○	○
datetimeoffset	●	●	●	●	●	●	●	●	●	●		●	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	●	○	○	○
datetime2	●	●	●	●	●	●	●	●	●	●	●		○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	●	○	○	○
decimal	●	●	●	●	●	●	●	●	○	○	○	○	✳	✳	●	●	●	●	●	●	●	●	●	●	○	○	○	○	●	○	○	○
numeric	●	●	●	●	●	●	●	●	○	○	○	○	✳	✳	●	●	●	●	●	●	●	●	●	●	○	○	○	○	●	○	○	○
float	●	●	●	●	●	●	●	●	○	○	○	○	●	●		●	●	●	●	●	●	●	●	●	○	○	○	○	●	○	○	○
real	●	●	●	●	●	●	●	●	○	○	○	○	●	●	●		●	●	●	●	●	●	●	●	○	○	○	○	●	○	○	○
bigint	●	●	●	●	●	●	●	●	○	○	○	○	●	●	●	●		●	●	●	●	●	●	●	○	○	○	○	●	○	○	○
int(INT4)	●	●	●	●	●	●	●	●	○	○	○	○	●	●	●	●	●		●	●	●	●	●	●	○	○	○	○	●	○	○	○
smallint(INT2)	●	●	●	●	●	●	●	●	○	○	○	○	●	●	●	●	●	●		●	●	●	●	●	○	○	○	○	●	○	○	○
tinyint(INT1)	●	●	●	●	●	●	●	●	○	○	○	○	●	●	●	●	●	●	●		●	●	●	●	○	○	○	○	●	○	○	○
money	●	●	●	●	●	●	●	●	○	○	○	○	●	●	●	●	●	●	●	●		●	●	●	○	○	○	○	●	○	○	○
smallmoney	●	●	●	●	●	●	●	●	○	○	○	○	●	●	●	●	●	●	●	●	●		●	●	○	○	○	○	●	○	○	○
bit	●	●	●	●	●	●	●	●	○	○	○	○	●	●	●	●	●	●	●	●	●	●		●	○	○	○	○	●	○	○	○
timestamp	●	●	●	●	○	○	●	●	○	○	○	○	●	●	●	●	●	●	●	●	●	●	●		○	●	○	○	○	○	○	○
uniqueidentifier	●	●	●	●	●	●	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○		○	○	○	●	○	○	○
image	●	●	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	●	○		○	○	○	○	○	○
ntext	○	○	●	●	●	●	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○		○	○	●	○	○
text	○	○	●	●	●	●	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	●		○	●	○	○
sql_variant	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	○	●	○	○	○		○	○	○
xml	●	●	●	●	●	●	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○		●	○
CLR UDT	●	●	●	●	●	●	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	●	○	○
hierarchyid	●	●	●	●	●	●	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○

● Explicit conversion
● Implicit conversion
○ Conversion not allowed

Example: Type Conversions

- Output the phrase Today's date is ##/##/####:

```
SELECT 'Today''s date is ' + GETDATE()
```

- Yield the following result

```
Msg 241, Level 16, State 1, Line 1
Conversion failed when converting date and/or time from
character string.
```

- Type conversion using 'CONVERT()'

```
SELECT 'Today''s date is ' +
       CONVERT(varchar(12), GETDATE(),101)
```

```
-----------------------------------
Today's date is 01/01/2012
(1 row(s) affected)
```

NULL Data

- The value of a row that doesn’t have any data for a particular column
- Values that are indeterminate are said to be NULL.
- By definition a NULL value means that we don’t know what the value is
  - The value could be 1, could be 347 or could be –294
  - It means we don’t know, undefined, or perhaps not applicable

Using SQL Server & Sample Database

SQL Server Identifiers

- The rules for naming in SQL Server are fairly relaxed, allowing things like embedded spaces and even keywords in names
- Like most freedoms, however, it's easy to make some bad choices and get yourself into trouble
- The name of the object must start with any letter, as defined by the specification for Unicode 3.2
  - Includes the letters most Westerners are used to: A–Z and a–z
  - "A" is different from "a" depends on the server configuration

SQL Server Identifiers

- The name of the object must start with any letter, as defined by the specification for Unicode 3.2. (cont.)
  - Includes the letters most Westerners are used to: A–Z and a–z.
  - “A” is different from “a” depends on the server configuration
  - After that first letter, almost any character can use
  - Up to 128 characters for normal objects and 116 for temporary objects
  - Any names that are the same as SQL Server keywords or contain embedded spaces must be enclosed in double quotes ("") or square brackets ([])

SQL Server Editions

- Data center
- Standard/Enterprise
- Developer
  - Full features with only development license
- Express
  - Offers database engine service and management tools for free

- http://www.microsoft.com/sqlserver/en/us/editions.aspx

SQL Server

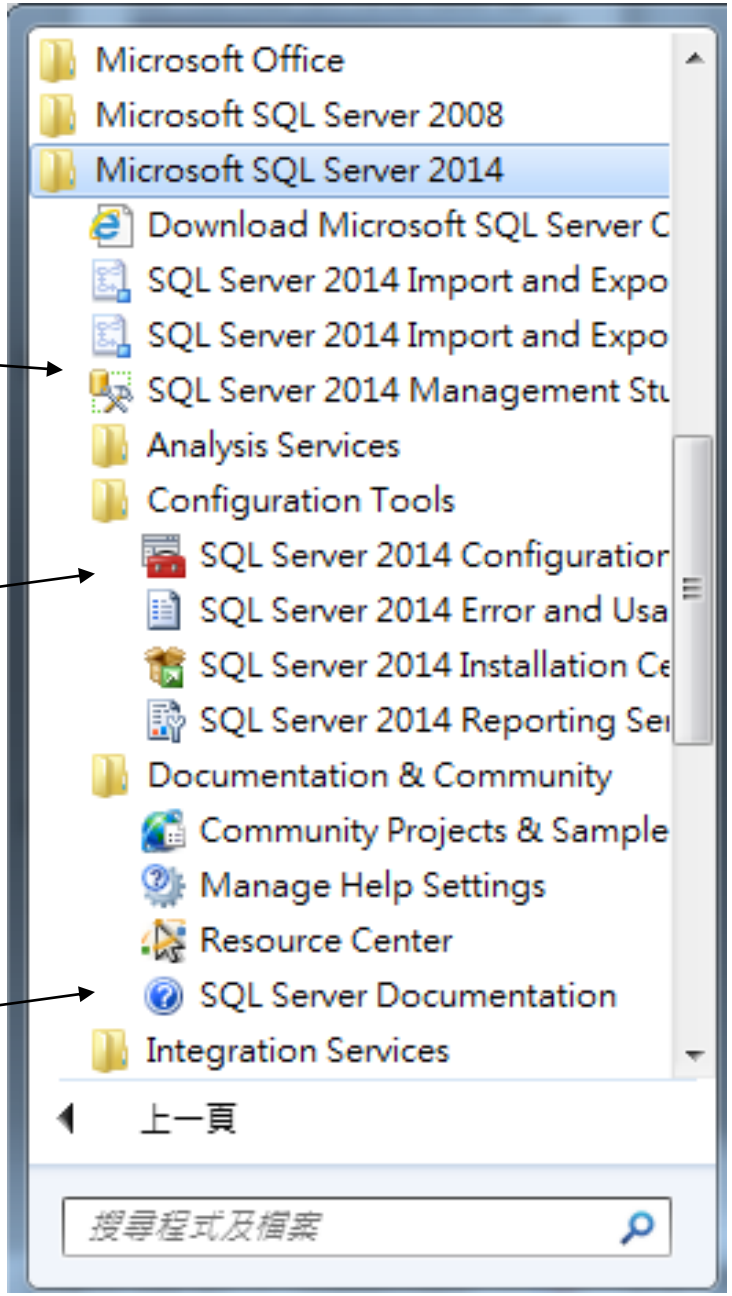
- Configuration manager
- SQL Server Management Studio
  - Login
  - Basic interface: object explorer and tool bar
- Database engine
  - Databases, tables, and other objects
  - Retrieving table data
  - Designing tables and columns
  - Editing tables and records
  - Relationship diagram
- Queries and views
  - Query editor
  - Running queries

Major Tools

Management Studio

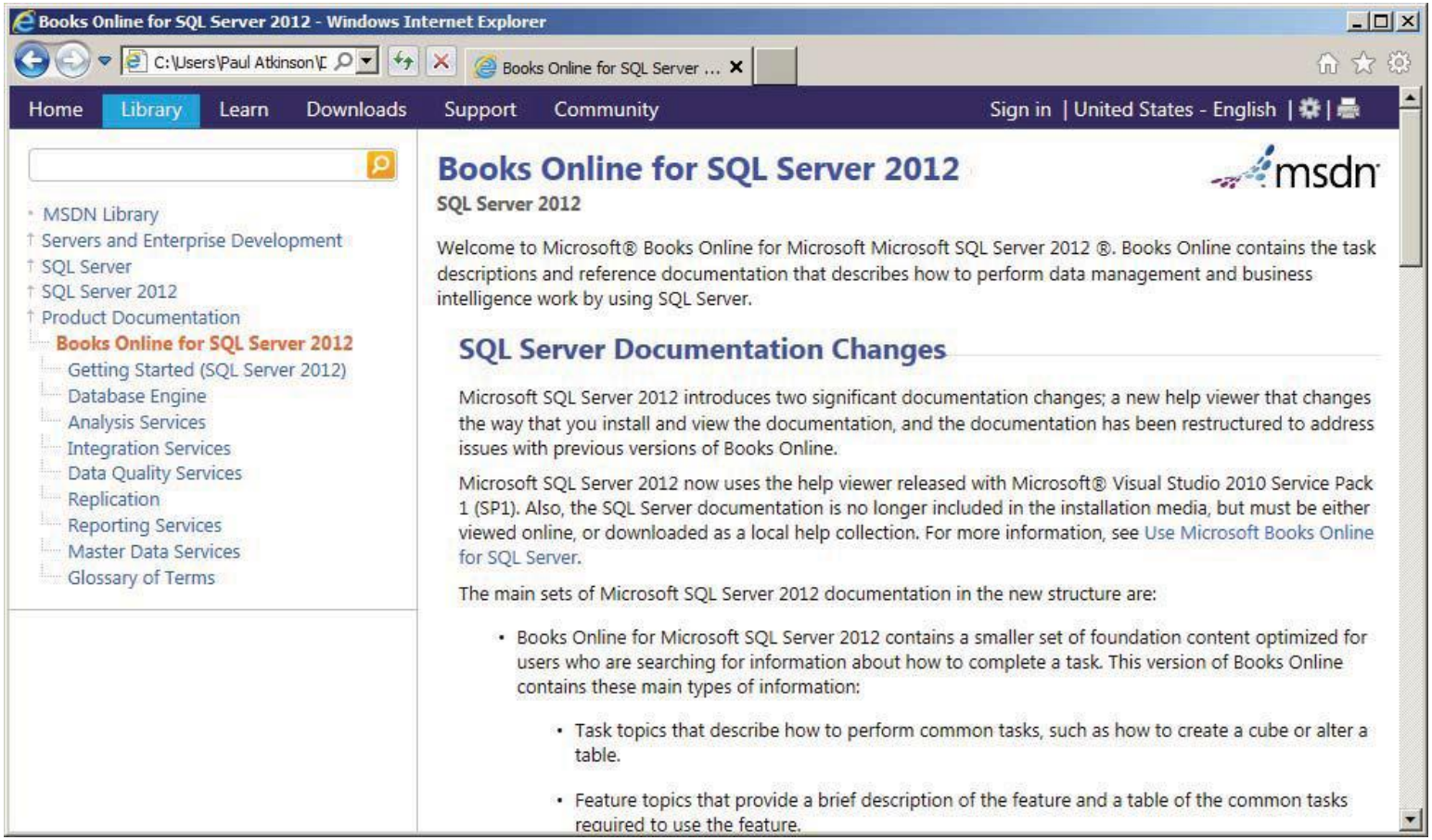
Configuration Manager

Books Online and Tutorials



Getting Help with Books Online

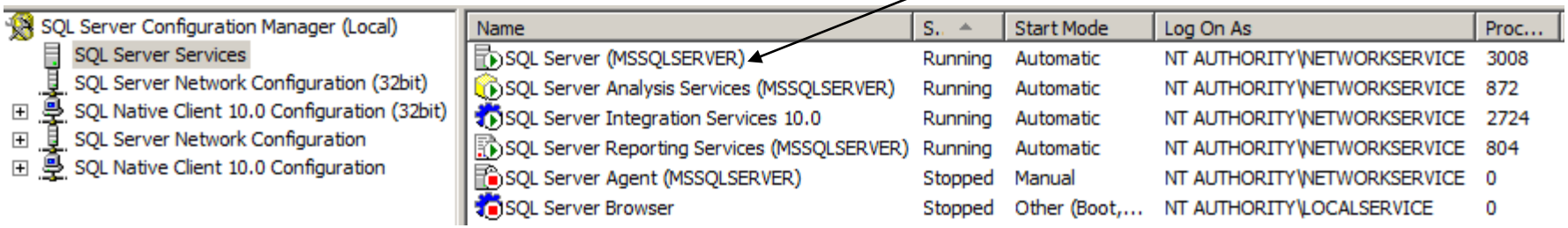
- Not going to remember everything
- Books Online is simply one of the most important tools

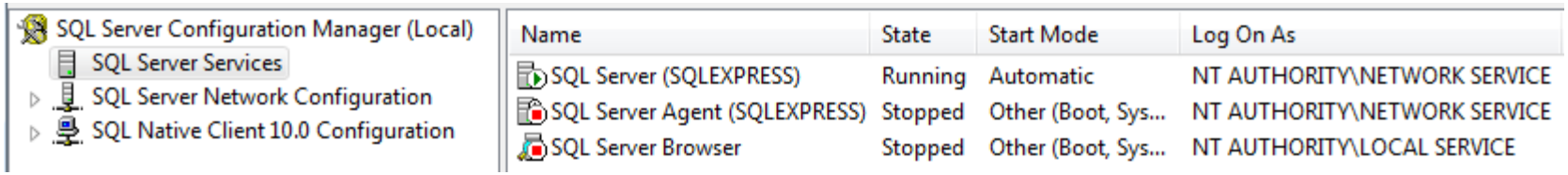


SQL Server Configuration Manager

- Configuration manager is a tool to view SQL Server service status, start or stop services
  - http://technet.microsoft.com/en-us/library/ms174212.aspx

Make sure the database server is running





SQL Server Management Studio

- The SQL Server Management Studio provides a variety of functionality for managing your server using a relatively easy-to-use graphical user interface
  - Create, edit, and delete databases and database objects
  - Query your database using T-SQL
  - Display current activity, such as who is logged on, what objects are locked, and from which
  - Manage security, including such items as roles, logins, and remote and linked servers
  - etc

SQL Server Management Studio

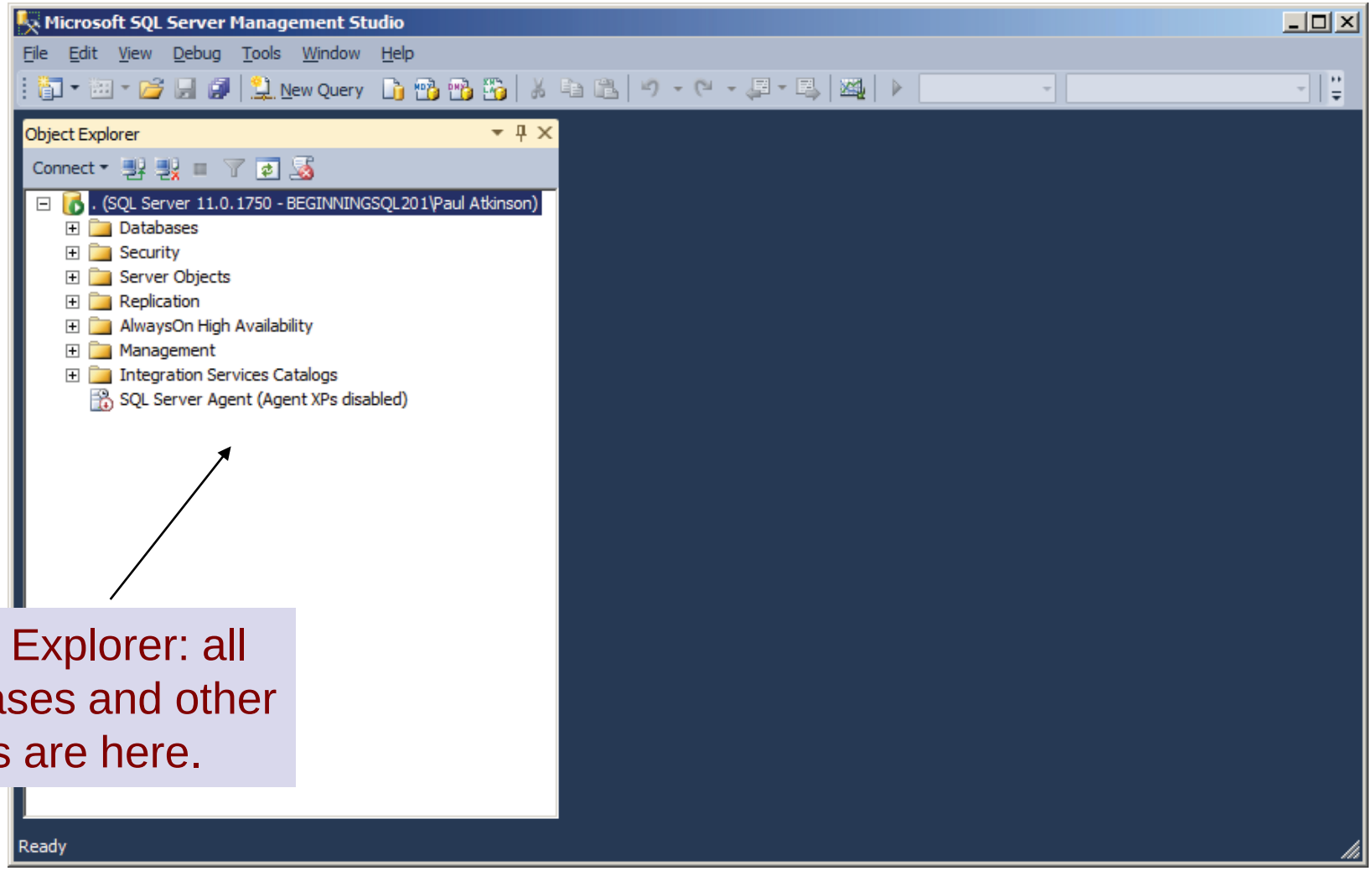
- Server Type: This relates to which of the various subsystems of SQL Server you are logging in to.
- Server Name: The SQL Server in to which you're asking to be logged.
- Authentication: Choose between Windows Authentication and SQL Server Authentication.

  - Windows Authentication: Windows users are mapped into SQL Server logins in their Windows user profile
  - SQL Server Authentication: The user provides a SQL Server – specific login and password

Making the Connection

- To log in the server, this will depend on whether the server was installed with Windows Authentication or Mixed Mode
- If installed with Mixed Mode, follow these steps to log in
  - Choose the (local) option for the SQL Server.
  - Select SQL Server Authentication.
  - Select a login name of **sa**, which stands for System Administrator. Alternatively, we may log in as a different user, as long as that user has system administrator privileges.
  - Enter the same password that we set when we installed SQL Server. On case-sensitive servers, the login is also case sensitive, so make sure we enter it in lowercase
  - Click Connect.

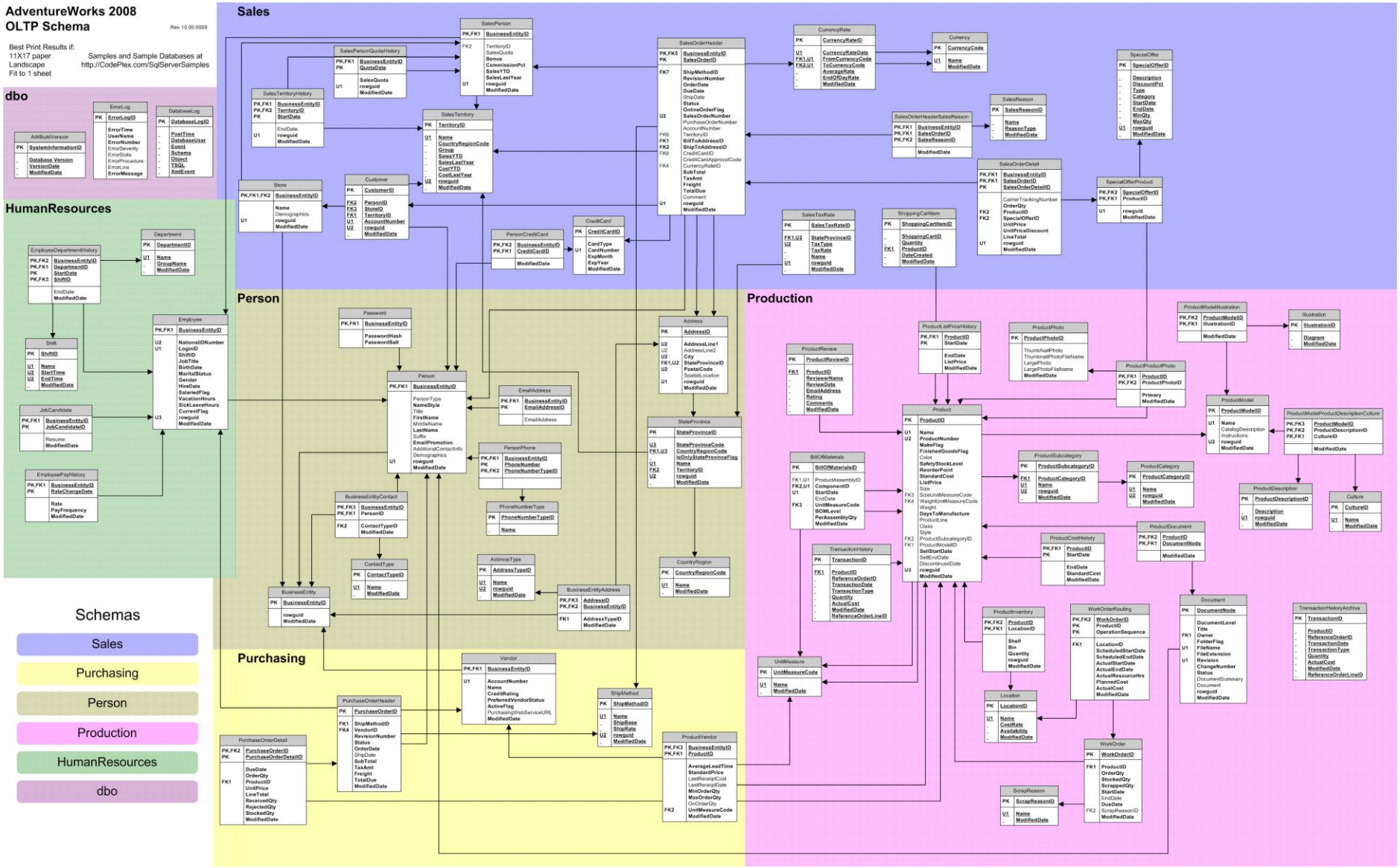
Object Explorer



Object Explorer: all databases and other objects are here.

AdventureWorks Database

- A far more robust sample database that would act as a sample for as much of the product as possible
- AdventureWorks Database
  - A fairly complete sample
  - Realistic volumes of data, complex structures, sections
  - Samples for the vast majority of product features

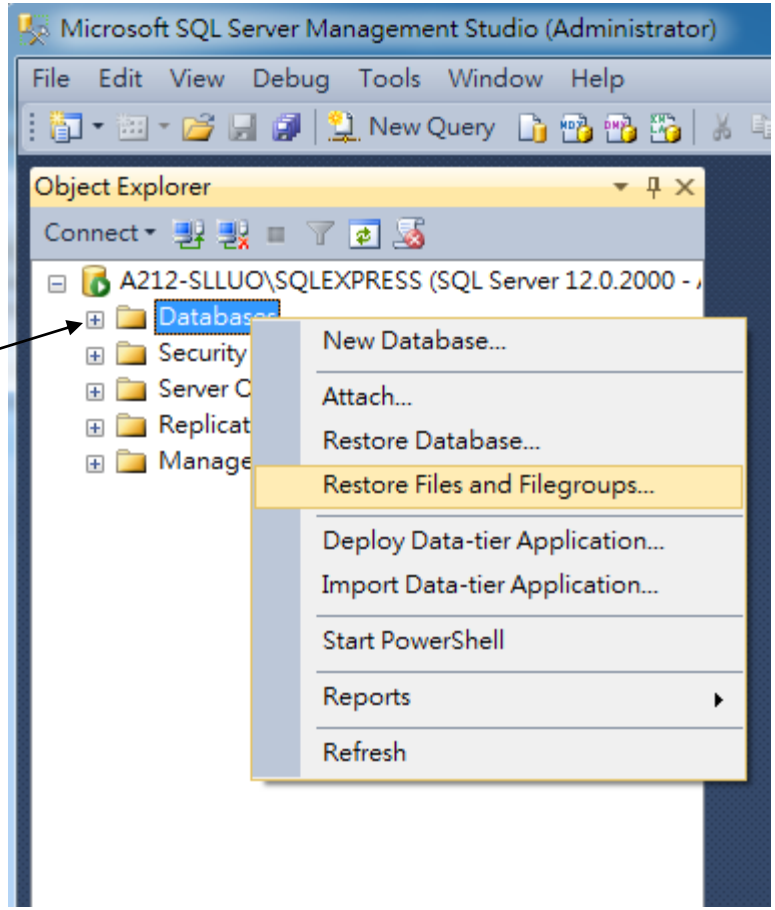

AdventureWorks 2008
OLTP Schema
Rev 10.00.0009
Best Print Results if:
11X17 paper
Landscape
Fit to 1 sheet
Samples and Sample Databases at
http://CodePlex.com/SqlServerSamples
dbo
HumanResources
Sales
Person
Production
Purchasing
Schemas
Sales
Purchasing
Person
Production
HumanResources
dbo

AdventureWorks Database

Schema	Contains objects related to	Examples
HumanResources	Employees of Adventure Works Cycles.	Employee Department
Person	Names and addresses of individual customers, vendors, and employees.	Contact Address StateProvince
Production	Products manufactured and sold by Adventure Works Cycles.	BillOfMaterials Product WorkOrder
Purchasing	Vendors from who parts and products are purchased.	PurchaseOrderDetail PurchaseOrderHeader Vendor
Sales	Customers and sales-related data.	Customer SalesOrderDetail SalesOrderHeader

Attach Sample Databases (1)

- Download the “Adventure Works” sample database and extract it.
  - AdventureWorks2014.bak

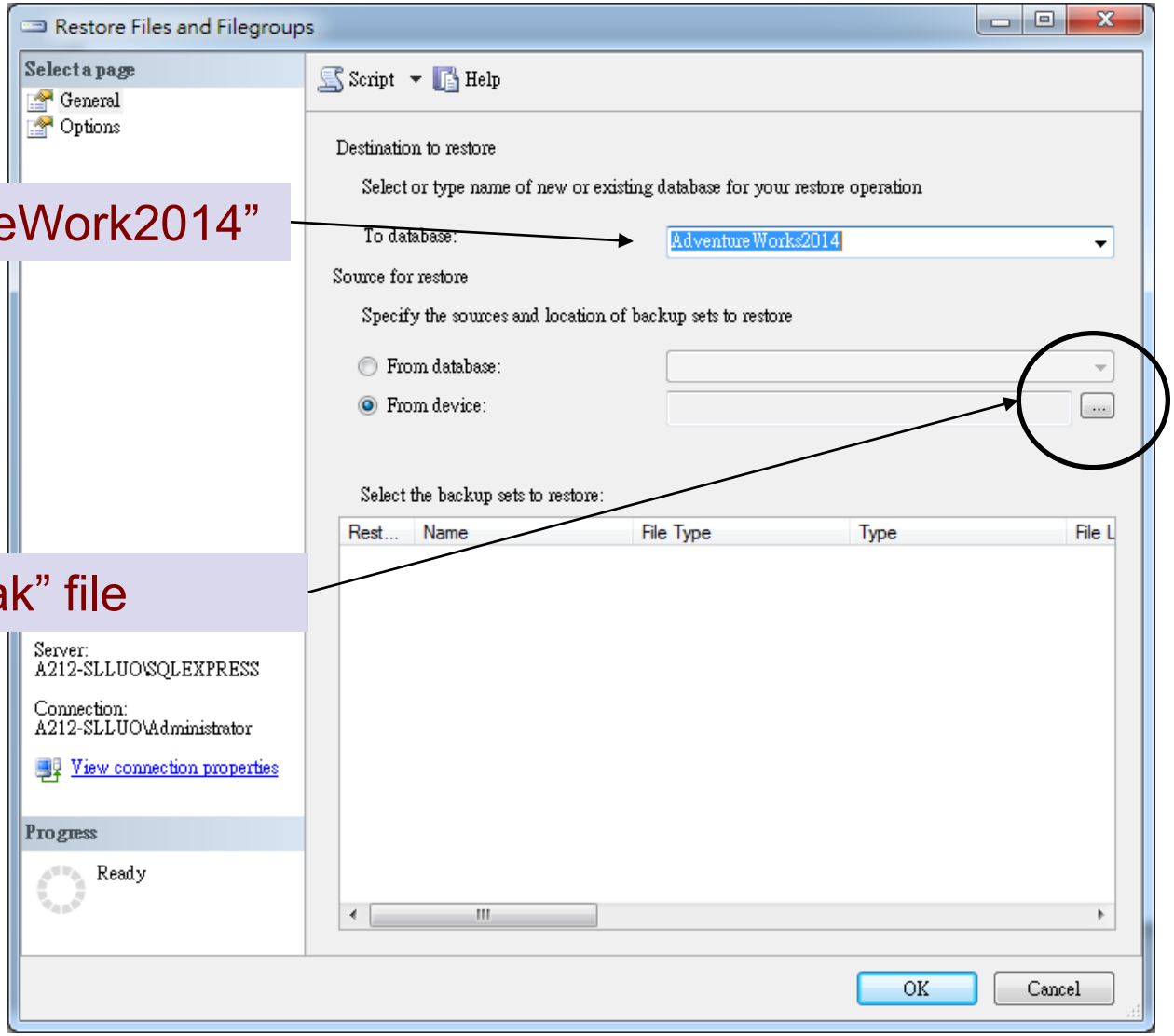


Right click on “Databases” to bring up the menu.

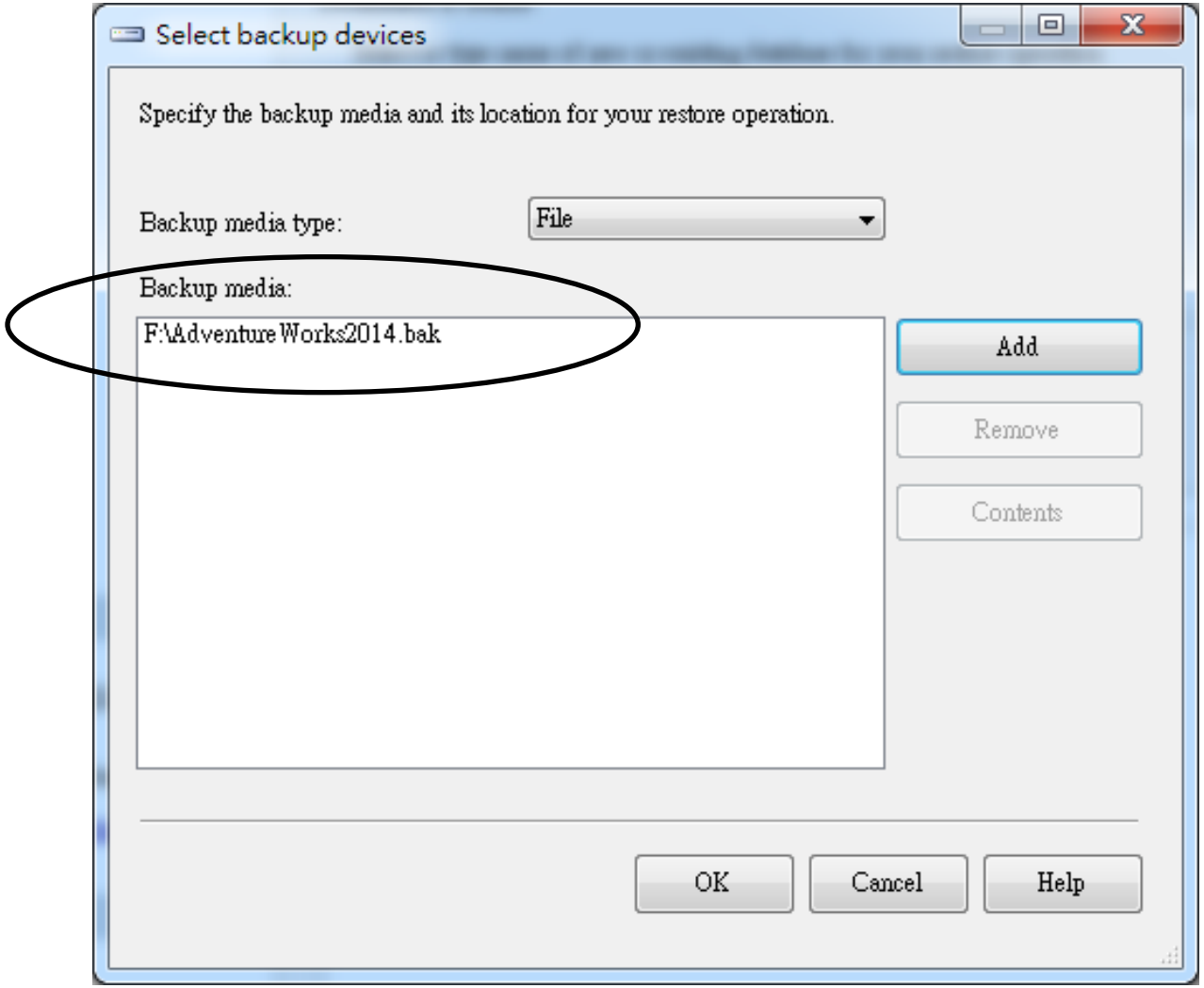
Attach Sample Databases (2)

Type “AdventureWork2014”

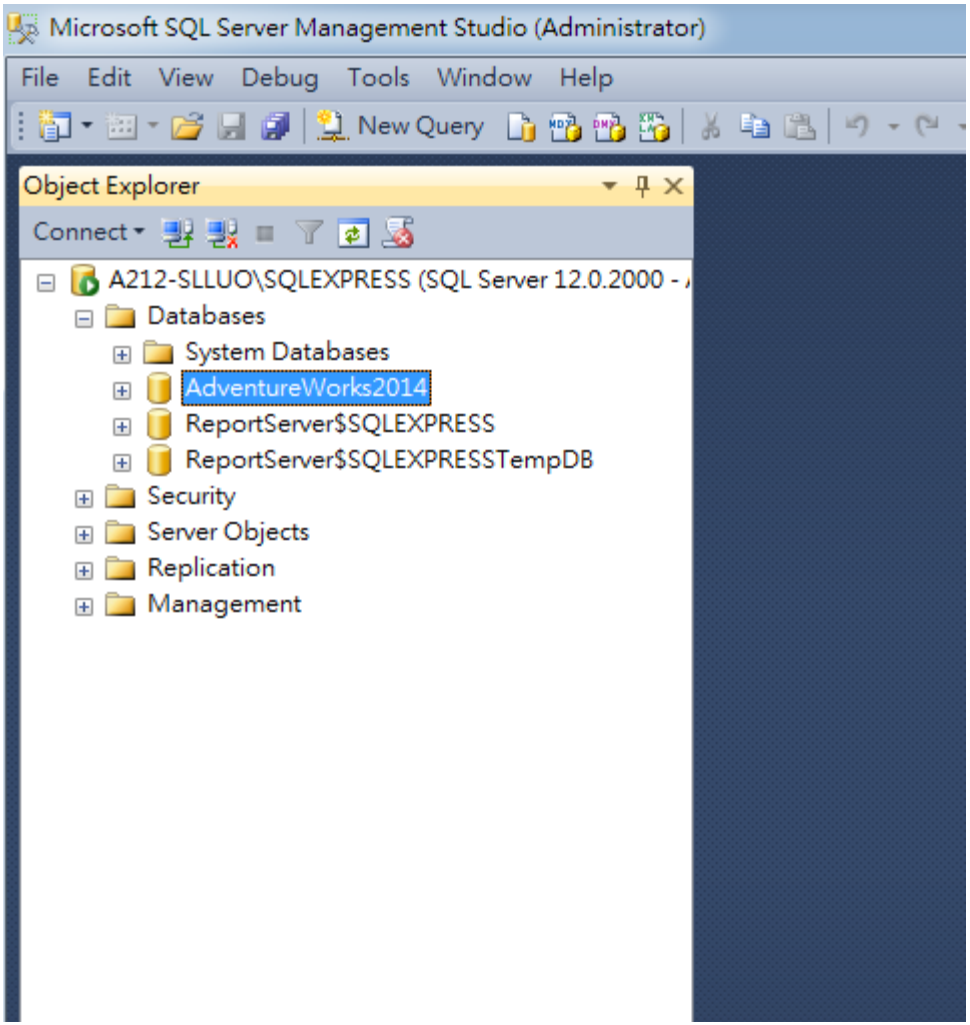
Select the “*.bak” file



Attach Sample Databases (3)



Attach Sample Databases (4)



Attach Sample Databases

- On the Standard toolbar, click the New Query button.
- Execute the following code in the query window:

```
USE [master]

RESTORE DATABASE AdventureWorks2014
FROM disk = 'C:\Program Files\Microsoft SQL
Server\MSSQL12.MSSQLSERVER\MSSQL\Backup\AdventureWorks2014.bak'
WITH MOVE 'AdventureWorks2014_data' TO 'C:\Program
Files\Microsoft SQL
Server\MSSQL12.MSSQLSERVER\MSSQL\DATA\AdventureWorks2014.mdf',
MOVE 'AdventureWorks2014_Log' TO 'C:\Program Files\Microsoft
SQL
Server\MSSQL12.MSSQLSERVER\MSSQL\DATA\AdventureWorks2014.ldf'
, REPLACE
```

Table Data

SQLQuery1.sql - A212-SLLUO\SQLEXPRESS.master (A212-SLLUO\Administrator (53)) - Microsoft SQL Server Management Studio (Administrator)

File Edit View Project Debug Tools Window Help

New Query

master | Execute | Debug

Object Explorer

Connect

- dbo.AWBuildVersion
- dbo.DatabaseLog
- dbo.ErrorLog
- HumanResources.Department
- HumanResources.Employee
- HumanResources.EmployeeDepart
- HumanResources.EmployeePayHist
- HumanResources.JobCandidate
- HumanResources.Shift
- Person.Address
- Person.AddressType
- Person.BusinessEntity
- Person.BusinessEntityAddress
- Person.BusinessEntityContact
- Person.ContactType
- Person.CountryRegion
- Person.EmailAddress
- Person.Password
- Person.Person
- Person.PersonPhone
- Person.PhoneNumberType
- Person.StateProvince
- Production.BillOfMaterials
- Production.Culture
- Production.Document
- Production.Illustration
- Production.Location
- Production.Product

SQLQuery1.sql - A2...dministrator (53))

```
/****** Script for SelectTopNRows command from SSMS  ******/
SELECT TOP 1000 [BusinessEntityID]
      ,[PersonType]
      ,[NameStyle]
      ,[Title]
      ,[FirstName]
      ,[MiddleName]
      ,[LastName]
      ,[Suffix]
      ,[EmailPromotion]
      ,[AdditionalContactInfo]
      ,[Demographics]
      ,[rowguid]
      ,[ModifiedDate]
  FROM [AdventureWorks2014].[Person].[Person]
```

Right click for menus

100 %

Results | Messages

	BusinessEntityID	PersonType	NameStyle	Title	FirstName	MiddleName	LastName	Suffix	EmailPromotion	A
1	1	EM	0	NULL	Ken	J	Sánchez	NULL	0	N
2	2	EM	0	NULL	Terri	Lee	Duffy	NULL	1	N
3	3	EM	0	NULL	Roberto	NULL	Tamburello	NULL	0	N
4	4	EM	0	NULL	Rob	NULL	Walters	NULL	0	N
5	5	EM	0	Ms.	Gail	A	Erickson	NULL	0	N
6	6	EM	0	Mr.	Jossef	H	Goldberg	NULL	0	N
7	7	EM	0	NULL	Dylan	A	Miller	NULL	2	N
8	8	EM	0	NULL	Diane	L	Margheim	NULL	0	N

Query executed successfully. | A212-SLLUO\SQLEXPRESS (12.0... | A212-SLLUO\Administrat... | master

Ready

Database Diagram

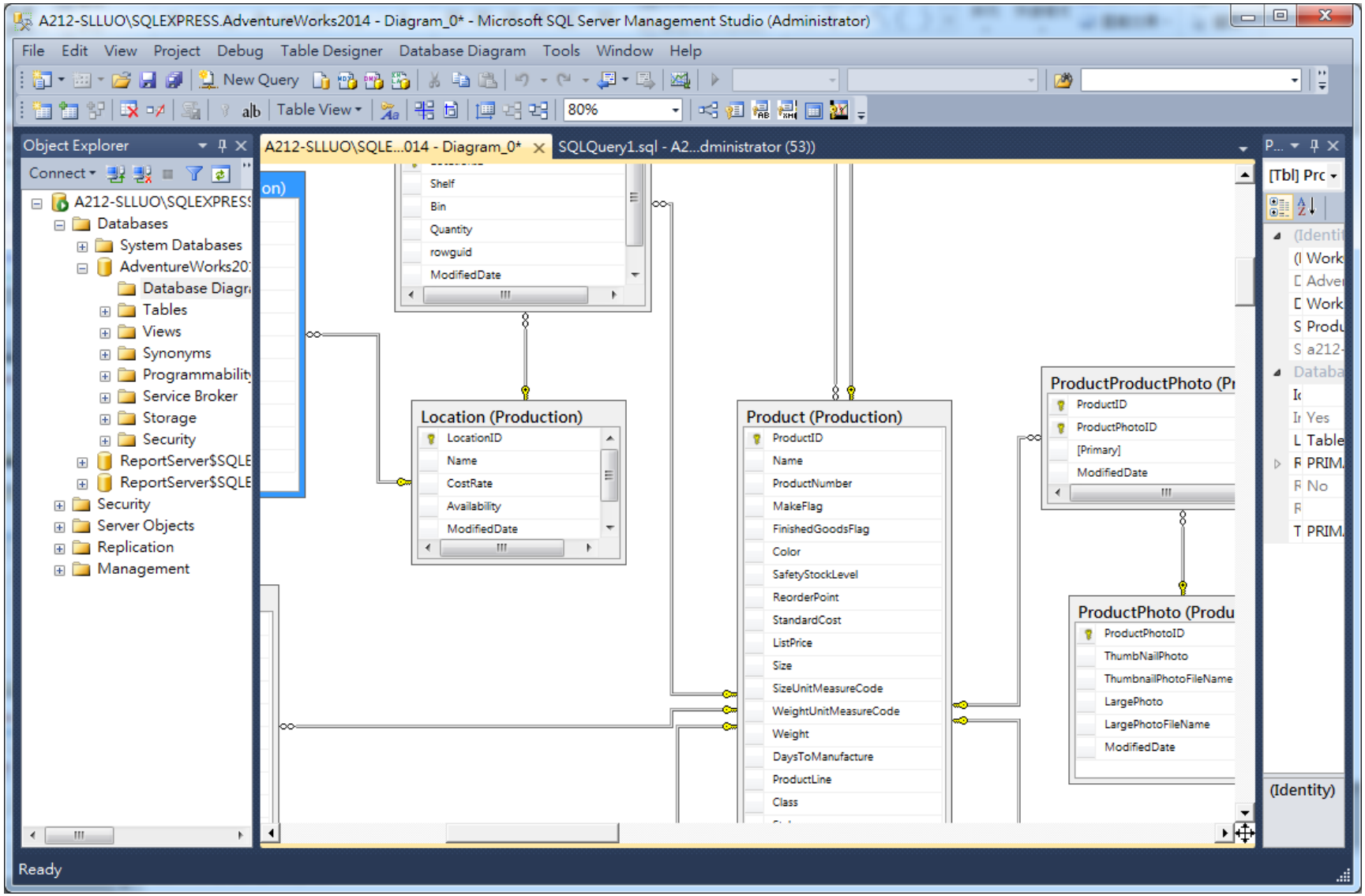


Table Design Metadata

A212-SLLUO\SQLEXPRESS.AdventureWorks2014 - Person.Person - Microsoft SQL Server Management Studio (Administrator)

File Edit View Project Debug Table Designer Tools Window Help

New Query

Object Explorer

Connect

- Tables
  - System Tables
  - FileTables
  - dbo.AWBuildVersion
  - dbo.DatabaseLog
  - dbo.ErrorLog
  - HumanResources.Depa
  - HumanResources.Empl
  - HumanResources.Empl
  - HumanResources.Empl
  - HumanResources.JobC
  - HumanResources.Shift
  - Person.Address
  - Person.AddressType
  - Person.BusinessEntity
  - Person.BusinessEntityAc
  - Person.BusinessEntityCc
  - Person.ContactType
  - Person.CountryRegion
  - Person.EmailAddress
  - Person.Passwor
  - Person.Person
  - Person.PersonPhone
  - Person.PhoneNumberT
  - Person.StateProvince
  - Production.BillOfMateri
  - Production.Culture
  - Production.Document
  - Production.Illustration

A212-SLLUO\SQLEX...- Person.Person | A212-SLLUO\SQLE...014 - Diagram_0* | SQLQuery1.sql - A2...dministrator (53))

Column Name	Data Type	Allow Nulls
BusinessEntityID	int	☐
PersonType	nchar(2)	☐
NameStyle	NameStyle:bit	☐
Title	nvarchar(8)	☑
FirstName	Name:nvarchar(50)	☐
MiddleName	Name:nvarchar(50)	☑
LastName	Name:nvarchar(50)	☐
Suffix	nvarchar(10)	☑
EmailPromotion	int	☐
AdditionalContactInfo		☐
Demographics		☐
rowguid		☐
ModifiedDate		☐

Right click on a table and select “Design”

Column Properties

(General)	
(Name)	BusinessEntityID
Allow Nulls	No
Data Type	int
Default Value or Binding	

(General)

[Tbl] Per

(Identity)

Ready

Example: Query

New Query

SQLQuery2.sql - A212-SLLUO\SQLEXPRESS.AdventureWorks2014 (A212-SLLUO\Administrator (53))* - Microsoft SQL Server Management Studio (Administrator)

File Edit View Query Project Debug Tools Window Help

New Query | AdventureWorks2014 | Execute | Debug

Object Explorer

Connect

- Tables
  - System Tables
  - FileTables
  - dbo.AWBuildVersion
  - dbo.DatabaseLog
  - dbo.ErrorLog
  - HumanResources.Depa
  - HumanResources.Empl
  - HumanResources.Empl
  - HumanResources.Empl
  - HumanResources.JobC
  - HumanResources.Shift
  - Person.Address
  - Person.AddressType
  - Person.BusinessEntity
  - Person.BusinessEntityAc
  - Person.BusinessEntityCc
  - Person.ContactType
  - Person.CountryRegion
  - Person.EmailAddress
  - Person.Password
  - Person.Person
  - Person.PersonPhone
  - Person.PhoneNumberT
  - Person.StateProvince
  - Production.BillOfMateri
  - Production.Culture
  - Production.Document
  - Production.Illustration

SQLQuery2.sql - A2...dministrator (53))*

```
SELECT * FROM Person.Address
```

100 %

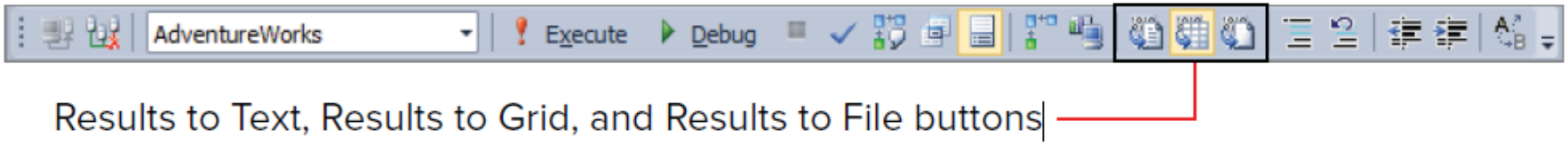
Results | Spatial results | Messages

	AddressID	AddressLine1	AddressLine2	City	StateProvinceID	PostalCode	SpatialLocation
1	1	1970 Napa Ct.	NULL	Bothell	79	98011	0xE6100000010CAE8BFC28BCE4474067A8
2	2	9833 Mt. Dias Blv.	NULL	Bothell	79	98011	0xE6100000010CD6FA851AE6D74740BC26
3	3	7484 Roundtree Drive	NULL	Bothell	79	98011	0xE6100000010C18E304C4ADE14740DA93
4	4	9539 Glenside Dr	NULL	Bothell	79	98011	0xE6100000010C813A0D5F9FDE474011A5C
5	5	1226 Shoe St.	NULL	Bothell	79	98011	0xE6100000010C61C64D8ABBD94740C460
6	6	1399 Firestone Drive	NULL	Bothell	79	98011	0xE6100000010CE0B4E50458DA47402F12A
7	7	5672 Hale Dr.	NULL	Bothell	79	98011	0xE6100000010C18E304C4ADE1474011A5C
8	8	6387 Scenic Avenue	NULL	Bothell	79	98011	0xE6100000010C0029A5D93BDF4740E2489
9	9	8713 Yosemite Ct.	NULL	Bothell	79	98011	0xE6100000010C6A80AD742DDC47408515

Query exe... | A212-SLLUO\SQLEXPRESS (12.0... | A212-SLLUO\Administrat... | AdventureWorks2014 | 00:00:00 | 19614 rows

Ready | Ln 1 | Col 29 | Ch 29 | INS

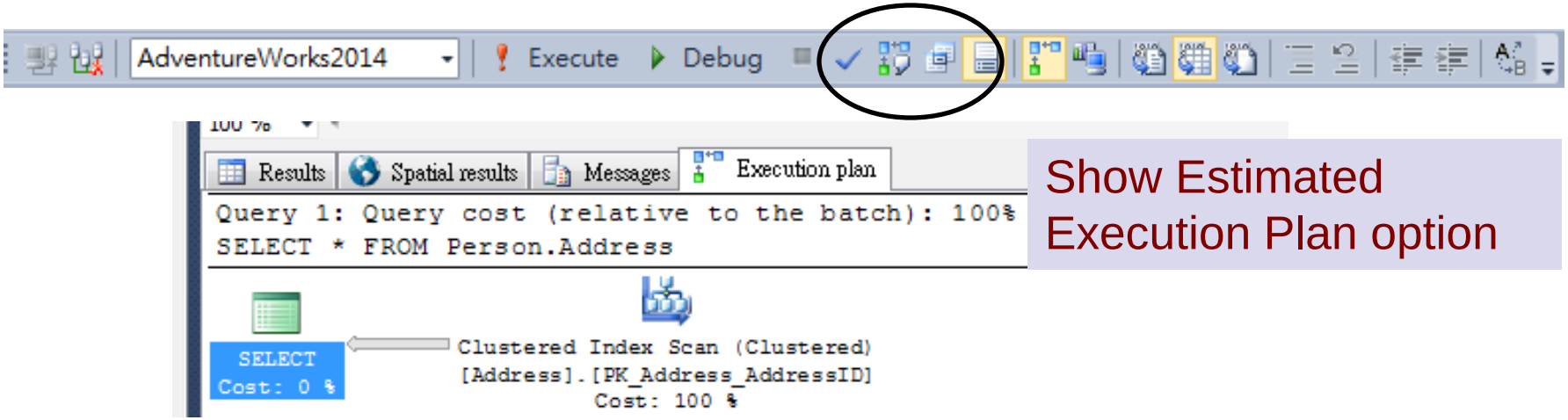
Example: Query Result



- Results to Text: The Results to Text option takes all the output from our query and puts it into one page of text results.
- Results to Grid: This option divides the columns and rows into a grid arrangement.
- Results to File: Think of this one as largely the same as Results to Text, but it routes the output directly to a file instead of to screen.

Example: Show Execution Plan

- Every time we run a query, SQL Server parses our query into its component parts and then sends it to the query optimizer.
- When we use the Show Estimated Execution Plan option, we receive a graphical representation and additional information about how SQL Server plans to run our query.



Summary

- SQL Server creates system databases and permits creation of many types of objects both within and external to those databases
- SQL Server provides a variety of data types that can be used to efficiently and correctly store information
- The objects names up to 128 characters (116 for temporary objects) that start with a letter