

**School of Information Technology**

Course : Diploma in Business Informatics

Subject : ITP282 - Enterprise Application Development & Project

AY / Sem : 2018 S2

Practical 5a: Review on Relational Database / Introduction to ADO.NET

OBJECTIVES:

By the end of this Practical students should be able to:

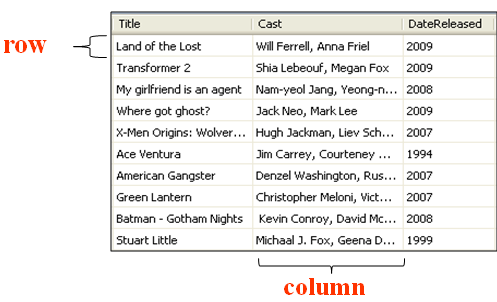
1. Create a database
2. Write SQL queries
3. Retrieve data from a Table using ADO.NET

**Practical 1: Review on Relational Databases / Introduction to ADO.NET**

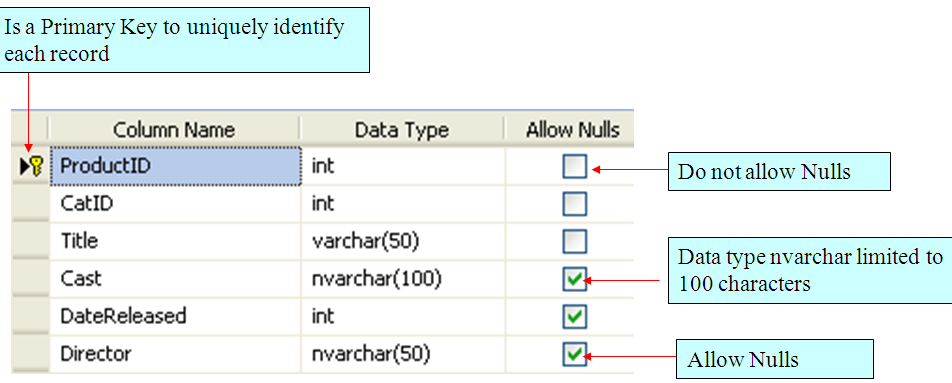
A database is a collection of data that is being organised centrally in a particular way that makes accessing to the data more efficiently. The most popular style of database system on personal computers is the relational database.

A database is composed of Tables. Any particular row of the table is called a record or a row;

Each column of the table represents a different field.



In a table, you would need to define properties and apply constraints to it to ensure validity of data. For example, the properties defined for the Product Table are:



In the above table,

* Column Name displays the name of that column
* Data Type refers to the data type of the column. In this example, only integer, nvarchar and varchar data type are used.
* Allow Nulls means that if you allow null values for a column, that column can be empty
* The key icon on the first column of the first row means that ProductID is a primary key. This is used to uniquely identify each record. For this column, it can also set another property – *Identity Column*. This will give it an auto incrementing value whenever a new record is created. By default, the starting value (Identity Seed) is 1 and auto increment (Identity Increment) is 1.

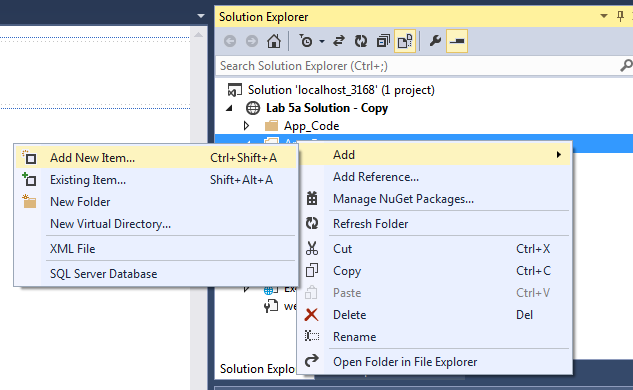
In the next exercise, you will create a database call DVDShop to manage and store data related to DVD movies. The relational database used will be SQL Server Express Edition (SSE). The advantage of using this database is as follows:

* + It is free download from Microsoft web site http://www.microsoft.com/express/sql/download/default.aspx
  + Automated connectivity to a SSE Instance. This means that database files can be treated as normal files allowing non admin Windows account to access SSE without extra privileges required

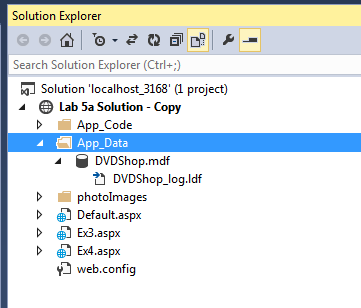
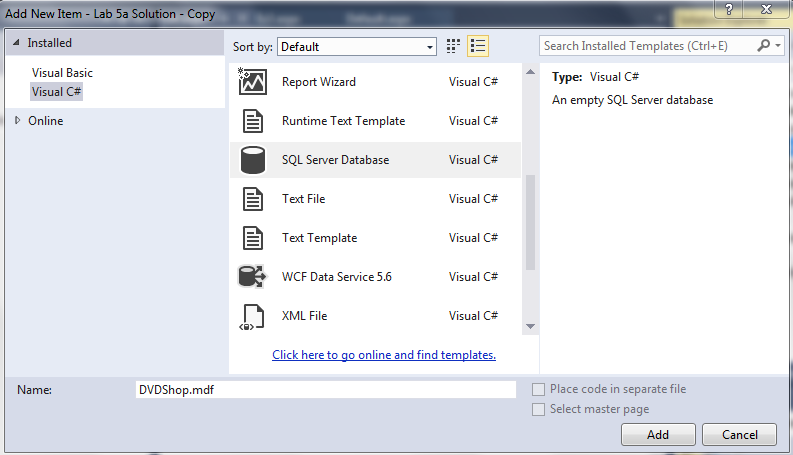
The database however is limited to 1 CPU and 4 GB RAM.

**Exercise 1: Creating Tables**

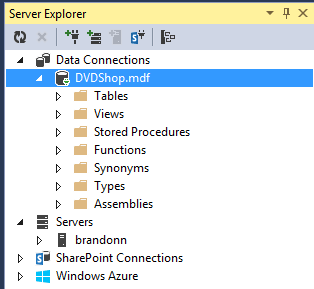
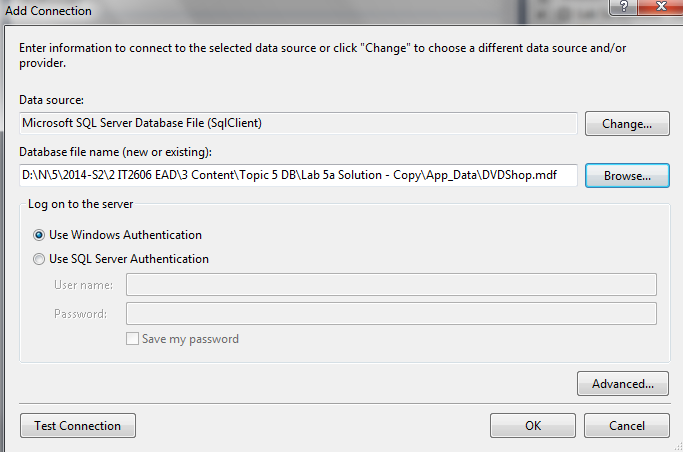
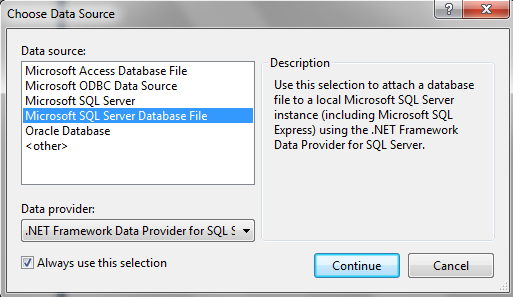
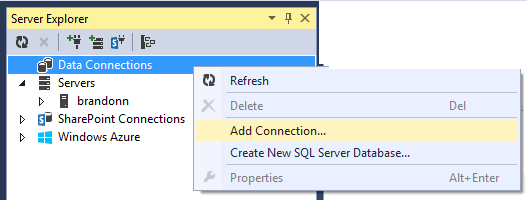
* 1. Create a new web site **Lab5a**.
  2. Right click on App\_Data folder and select Add new Item; Select SQL database in Add New Item dialog box.



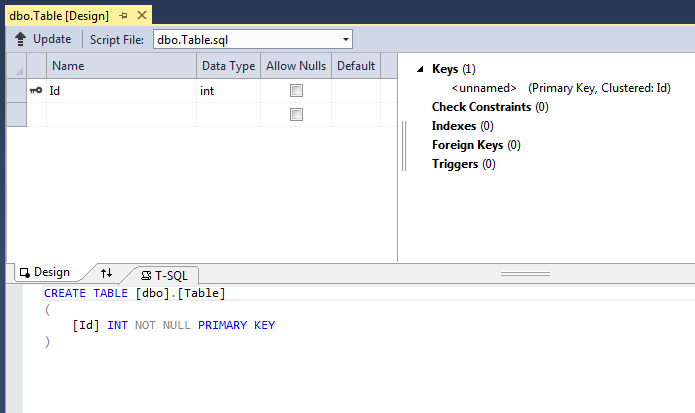
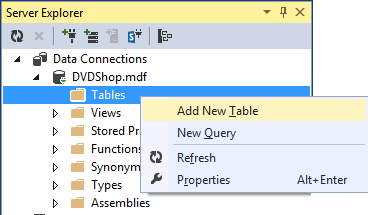
* 1. Name the database DVDShop.mdf.



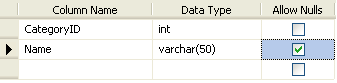
* 1. In Server Explorer, click on the Refresh icon. If the newly created database (DVDShop) appears, proceed to next step. Else, right click at **Data Connections** and **Add Connection**…(as following figures).



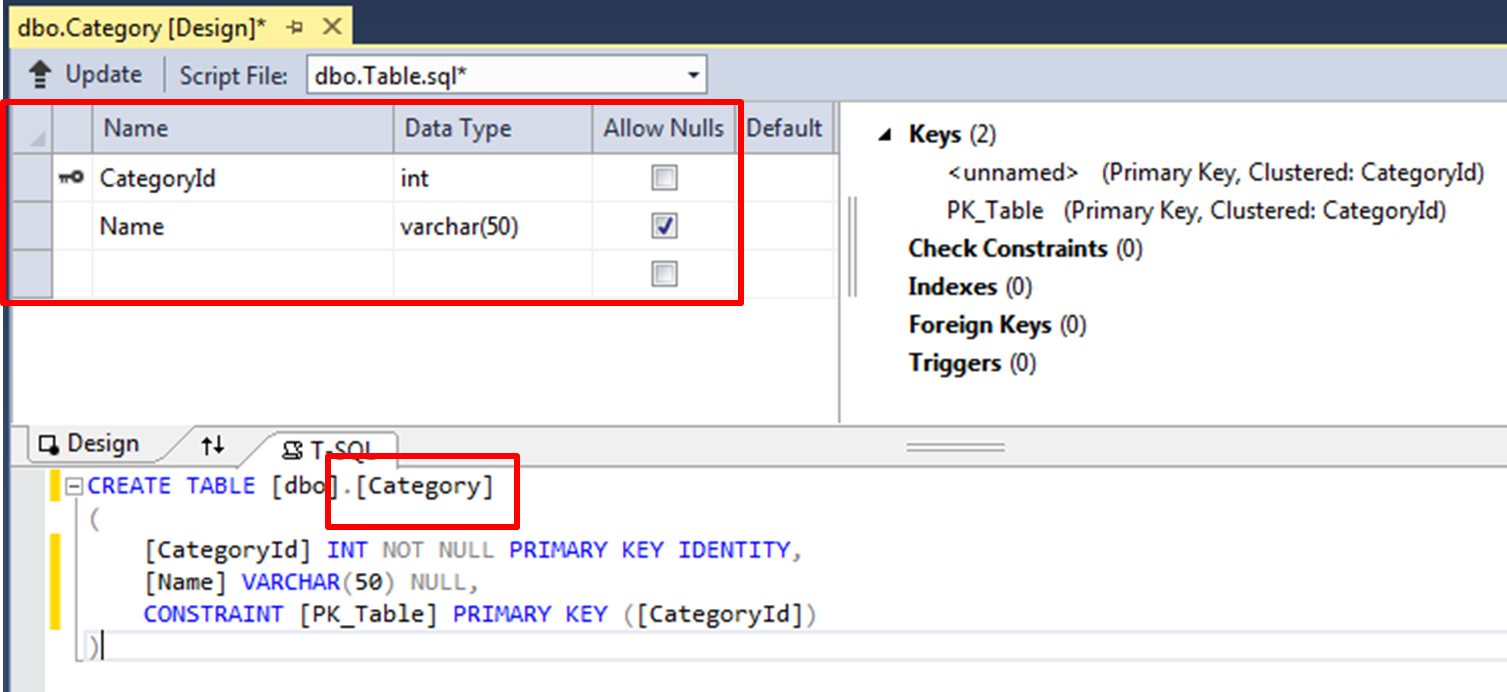
* 1. In the Server Explorer, right click Table and select Add New Table.



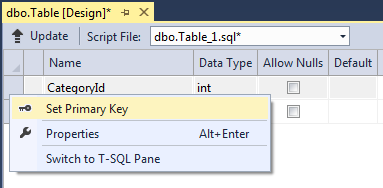
* 1. Enter the following values:



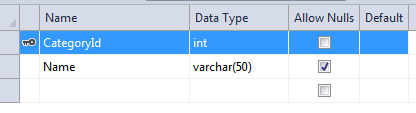
Name this new table as “Category”. The following figure is the outcome:



* 1. If CategoryID is not set as Primary Key, right click at the button next to CategoryID.



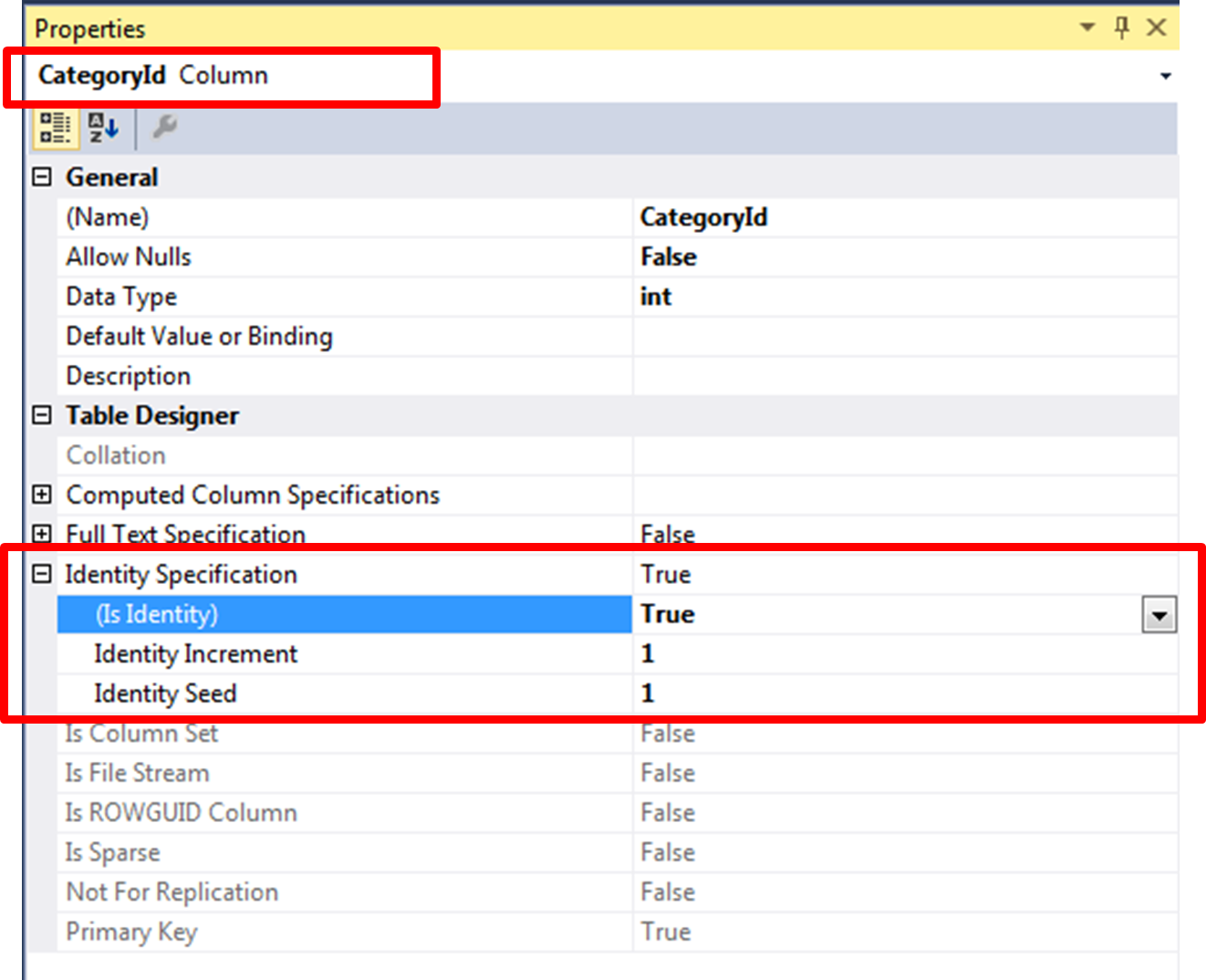
Select the CategoryID column.



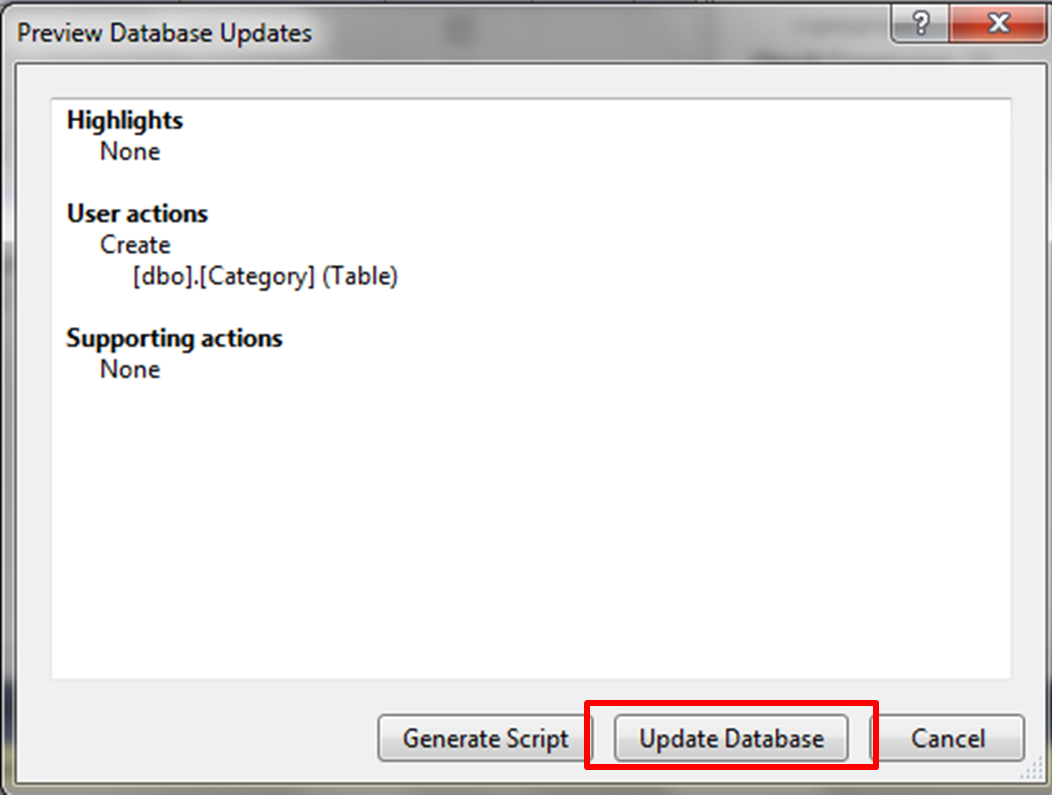
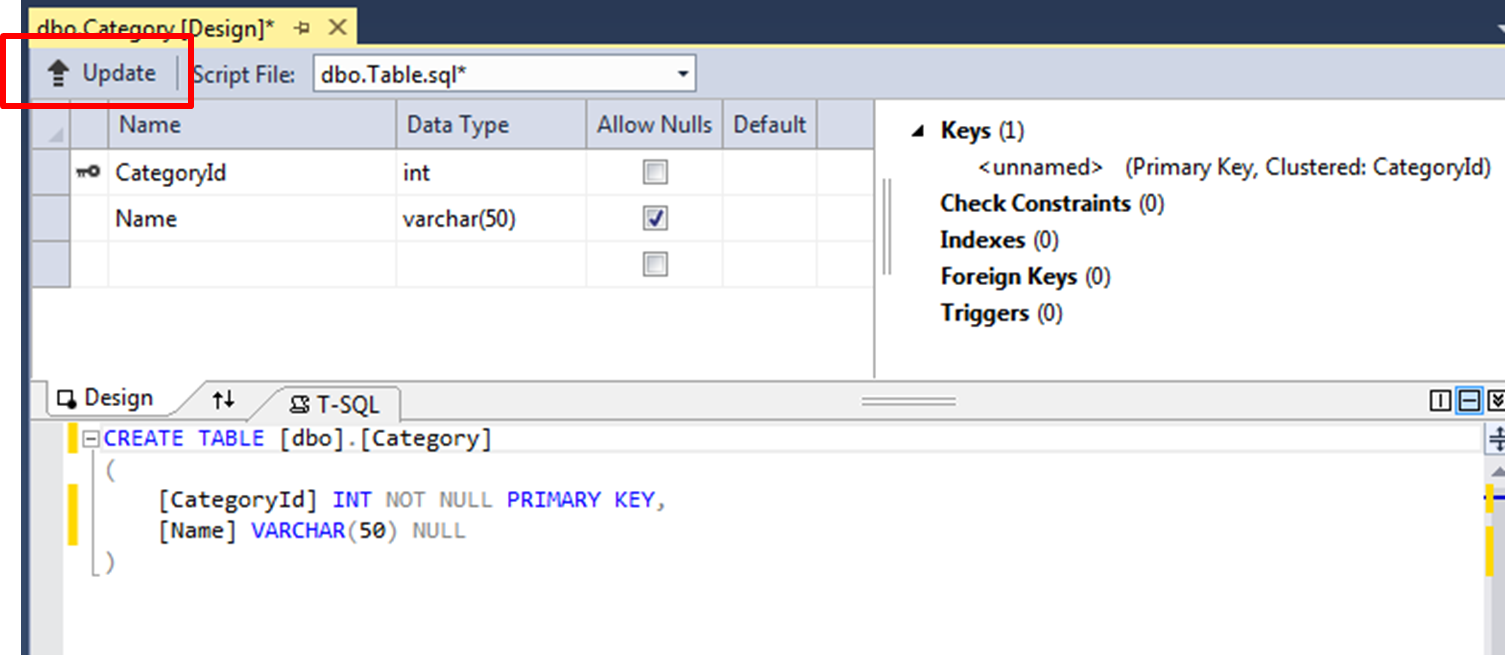
* 1. To make this column auto incrementing as well, we set its “Is identity” property to Yes. The default Identity Seed is 1 and Identity Increment is 1.

The default Identity Seed refers to the value given to first row created in the Table; The Identity Increment refers to the incremental value added to the current record to create the value for the next record of for that column.

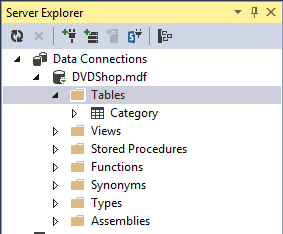
At Properties windows:



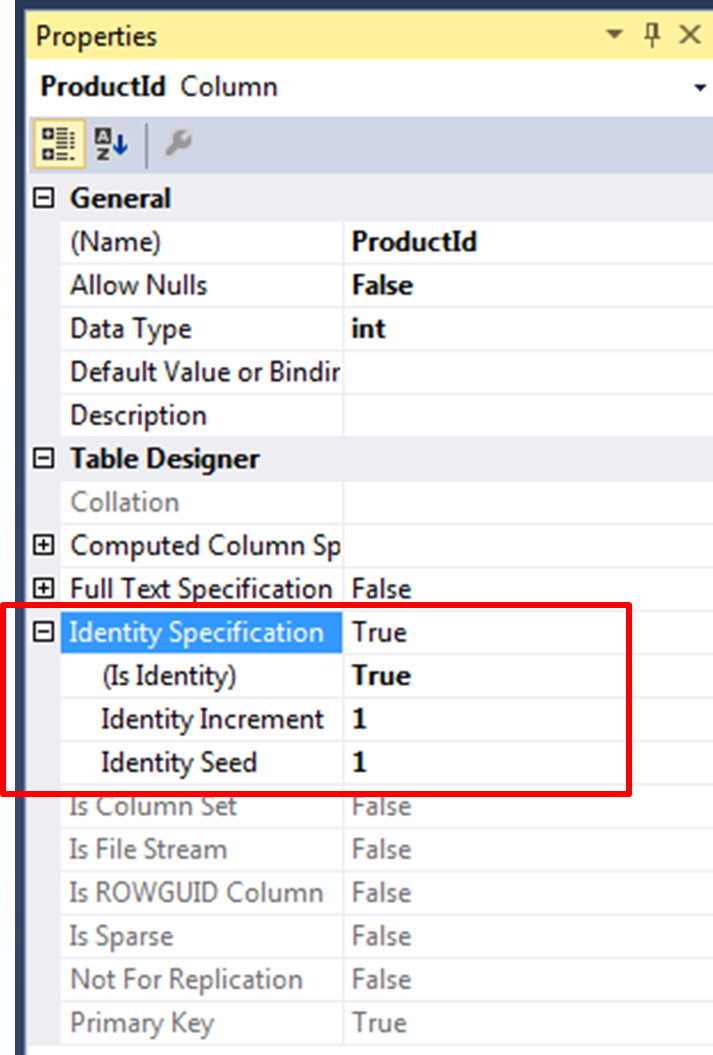
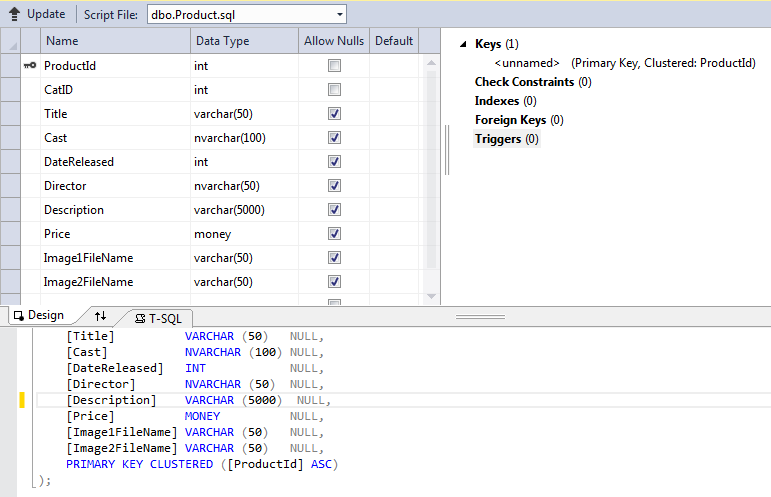
* 1. Save the table by clicking on the Update button (as following). This will trigger an Update effect to the newly created database (DVDShop.mdf).



* 1. Confirm that the table is created as displayed in the Server Explorer:

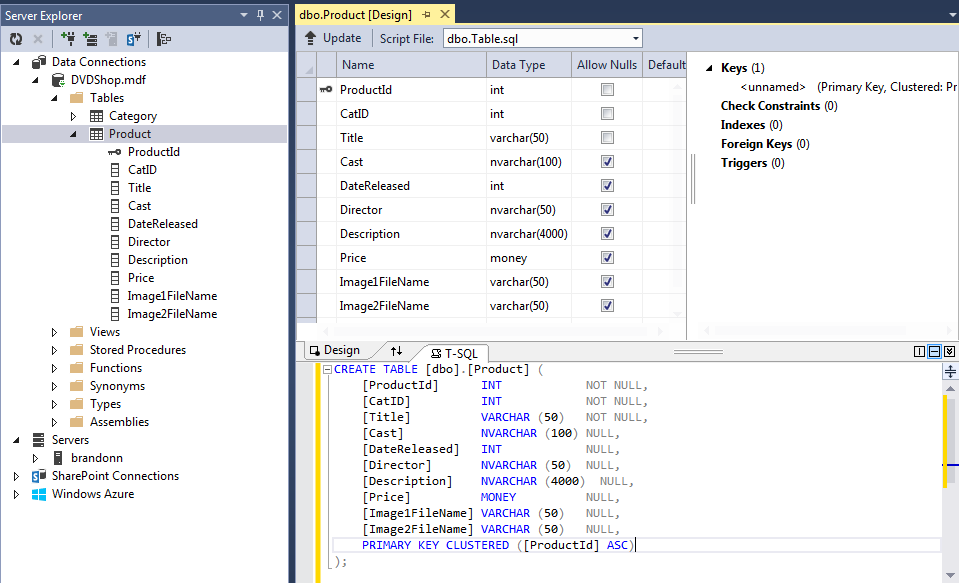


* 1. Create another Table named **Product** with the following columns. Make sure ProductID is defined as a Primary Key and IsIdentity= Yes.



* 1. To expedite the creation of data in both Category and Product Tables you have just created, follow the following steps to insert records from the category,sql and product.sql scripts:

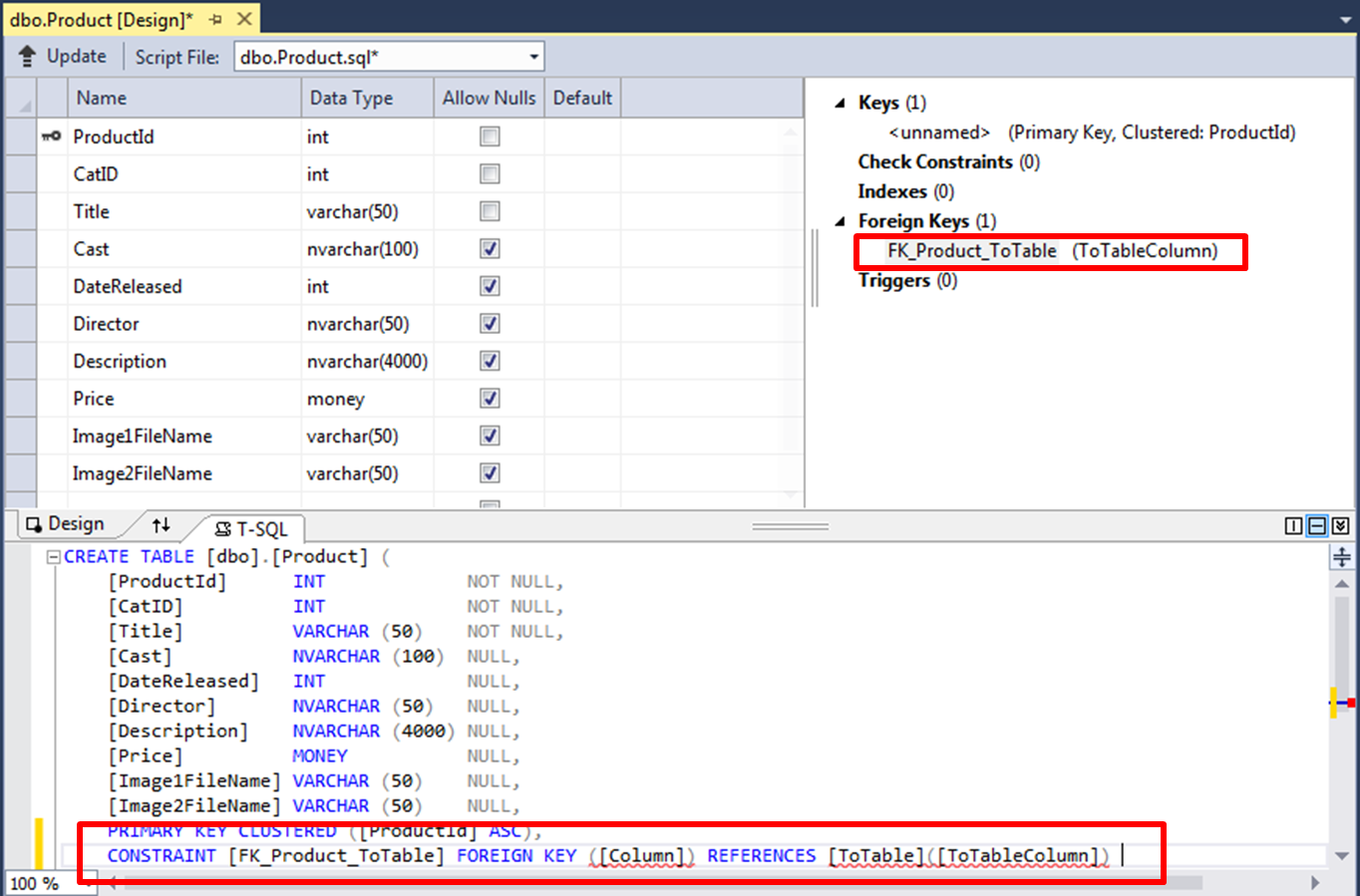
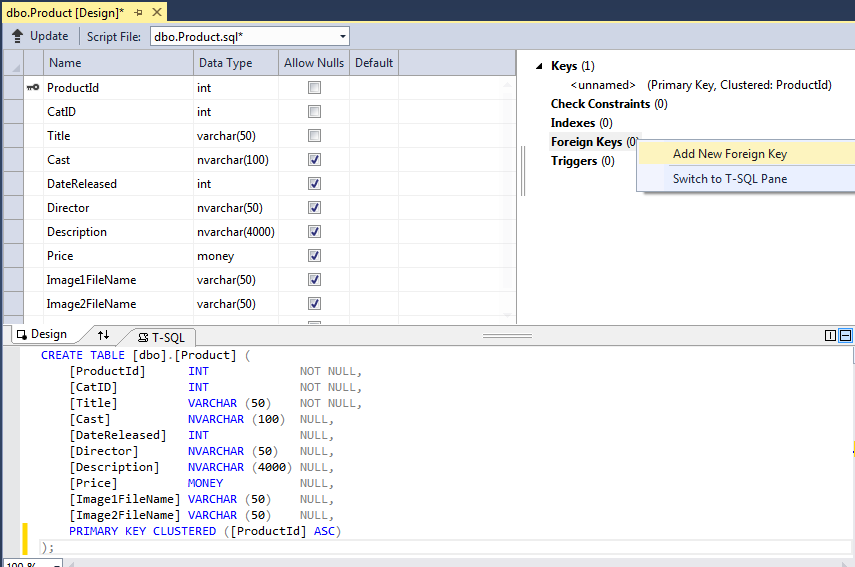
|  |
| --- |
| * Create a new SQL Query |
| * Copy and paste category.sql into the SQL Pane and Execute.  * Six records should be added to the Category Table. |
|  |
|  |
| * Repeat the steps above using Product.sql script to insert 20 records into the Product Table. |



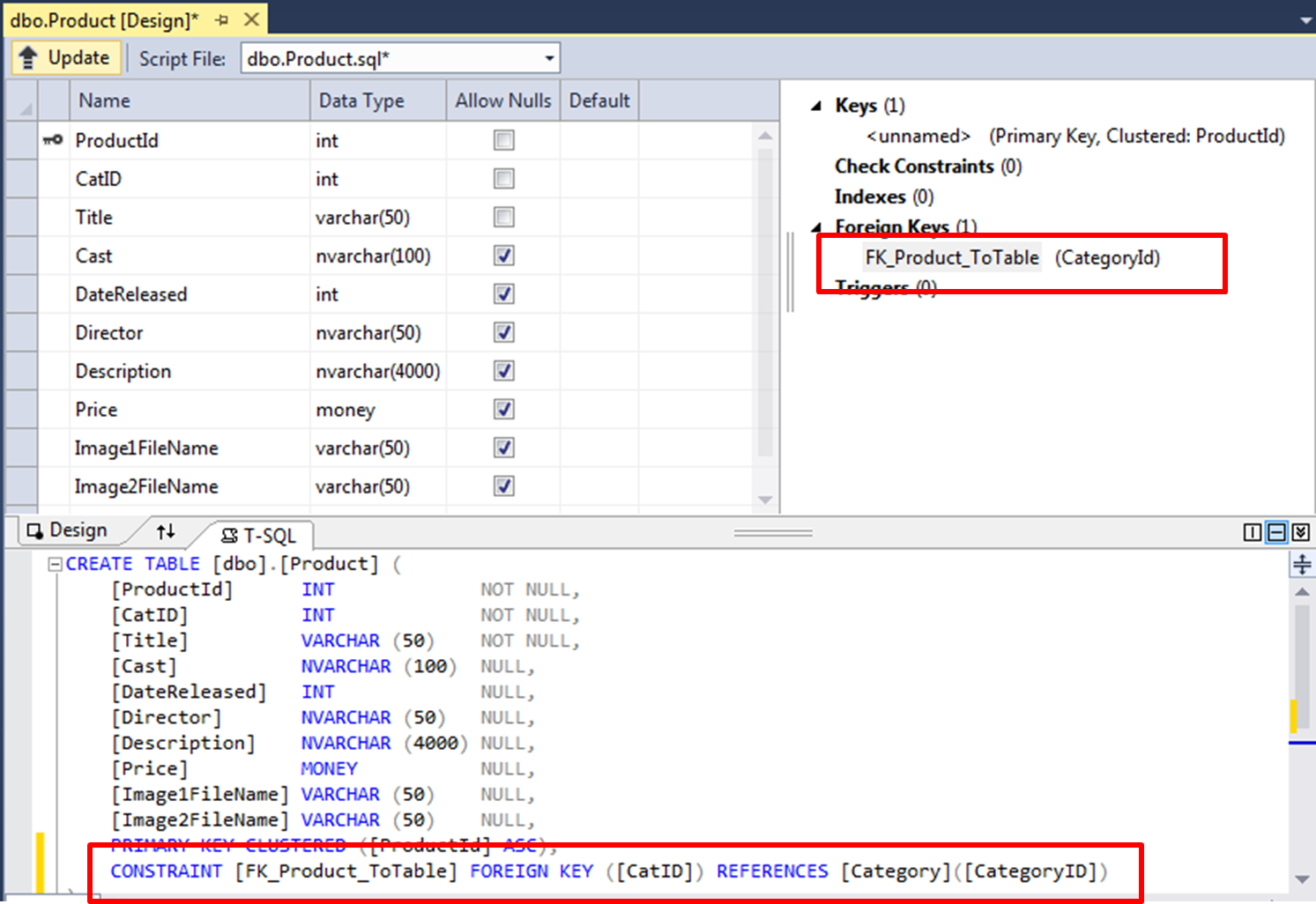
**Exercise 2: Creating relationships between Tables**

In the next exercise, you will create a One to Many relationships between the Category and Product Tables.

* 1. Right click on Foreign Keys (0) and click on Add New Foreign Key.



Amend the detail of the Constraint as following:-



**Exercise 3: SQL Fundamentals (Retrieving data)**

SQL is the international standard database language. Transact-SQL or T-SQL is the SQL dialect for SQL Server which will be used in this module.

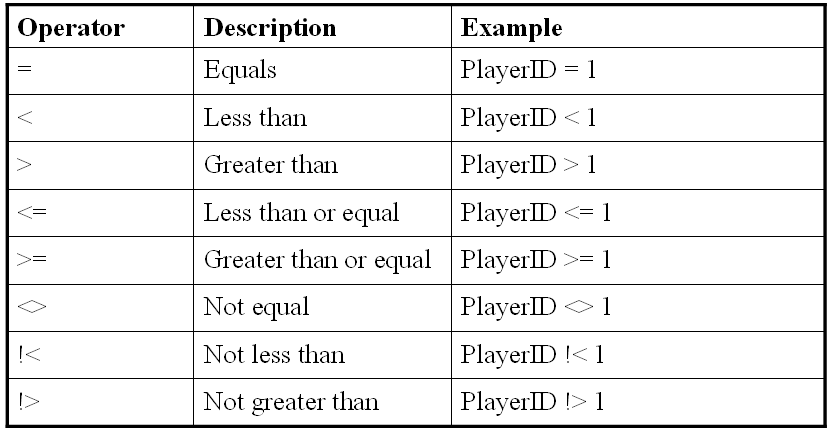
To execute SQL, tools are needed to pass the SQL statements to the database manager, including SQL Server Management Studio Express, SQLCMD ( An utility installed as part of the SQL Server installation) and Visual Studio Server Explorer.

Mastering SQL can be a subject by itself; for more information, refer to [www.w3cschools.com/sql](http://www.w3cschools.com/sql)

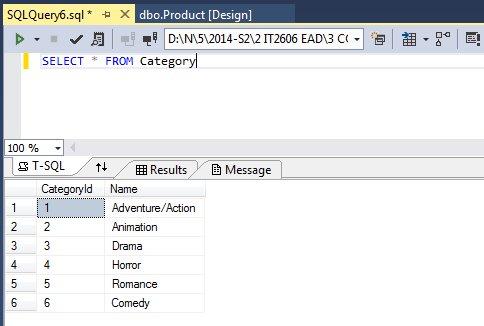
Below are some examples of simple SQL Statements and their expected results:

|  |  |
| --- | --- |
| SELECT \* FROM Category |  |
| SELECT Name FROM Category |  |
| SELECT CategoryID, Name FROM Category WHERE (CategoryID >= 3) |  |
| SELECT CategoryID, Name FROM Category WHERE (Name LIKE 'A%') |  |
| SELECT CategoryID, Name FROM Category WHERE (Name LIKE '[DH]%') |  |
| SELECT COUNT(\*) AS numberOfCategory FROM Category |  |

Additional operators can also be used in conjunction with the WHERE clause, for example,



* 1. In the Server Explorer, right click the Product Table and select New Query;



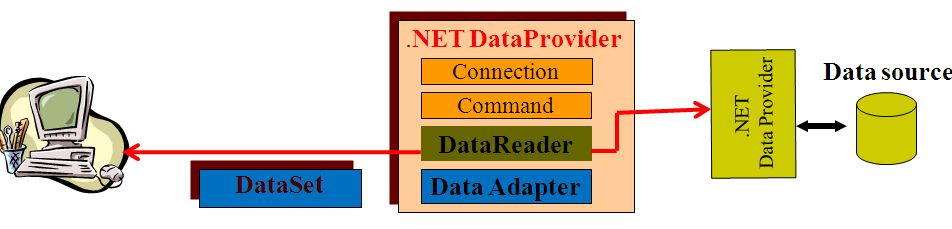
* 1. Translate the following requirements into SQL statements by typing the statements in the SQL Pane. To execute the query, right click and select !Execute SQL:
* Retrieve all records from Product Table. Display only results from columns ProductID, Title and Price.
* Retrieve all records from Product Table with Price greater than or equal to 18.50.
* Retrieve all records from Product Table but displaying only results from columns ProductID, Title and Price.
* Count the number of records in Product Table
* Retrieve all records from Product Table with first letter of Title starting with A. Display only results from columns ProductID, Title and Price.

**Exercise 4: Using Data Reader for database access**

ADO.NET supports 2 models for data access:

* **Disconnected model**: Data is retrieved using a Data Adapter class and stored in an in-memory DataSet. Connection to data source is then closed.
* **Connected model**: Data is read using a Data Reader class. Connection is kept open throughout the duration of data access.

In this module, we will be using primarily the connected model (data reader) for database retrieval:



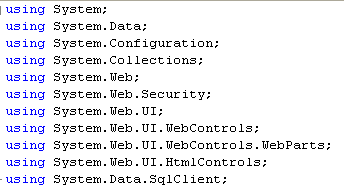
|  |
| --- |
|  |

Using Data Reader, the main steps for database retrieval are:

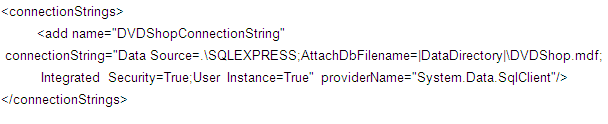
1. Define a connection to Database
2. Create a Command to retrieve data from Table in Database
3. Open Connection to database; Call command.ExecuteReader to retrieve data
4. Access data using DataReader
5. Close Reader and Connection

The following steps demonstrate how the above five steps is done using C #:

* 1. Using the same web site you created, create a new web page **Ex1.aspx**. Make sure System.Data.SqlClient namespace is included.



* 1. Make sure your database created earlier DVDShop.mdf is in App\_Data folder of your project .
  2. Create a web.config. Look for the ConnectionString element in web.config. Delete it and replace with the ConnectionString from **Lab5a\_ConnectionString.txt**. The name **DVDShopConnectionString**, as shown below, will be used by the C# program to retrieve the relevant connection information to the database.

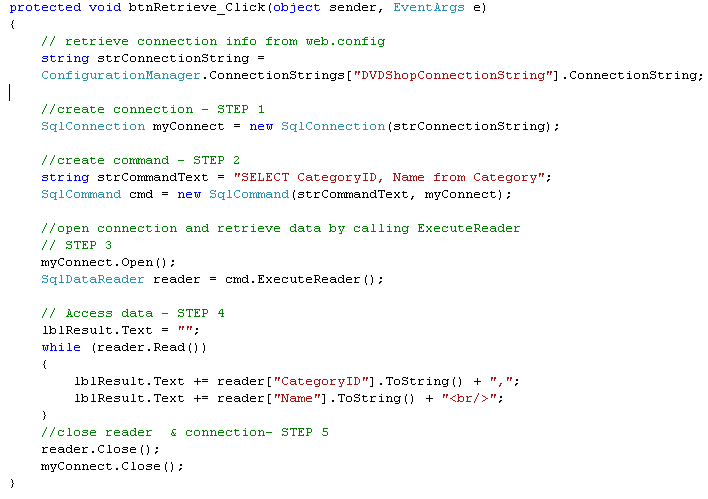


* 1. Create a Label with ID=lblResult.
  2. Create a button with ID=btnRetrieve and Text= Retrieve Category. Double click on it to go to the btnRetrieve\_Click event handler. Add in the following codes in the button click event hander. Notice the program comprises the five main steps shown earlier:

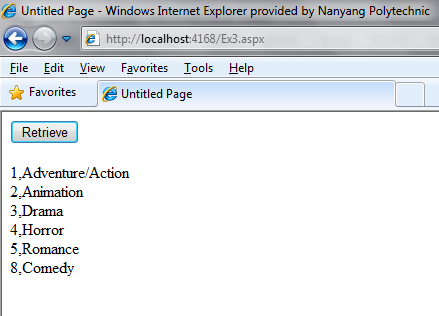
Keep looping till all data is read

Retrieve data from Name column

Retrieve data from CategoryID column

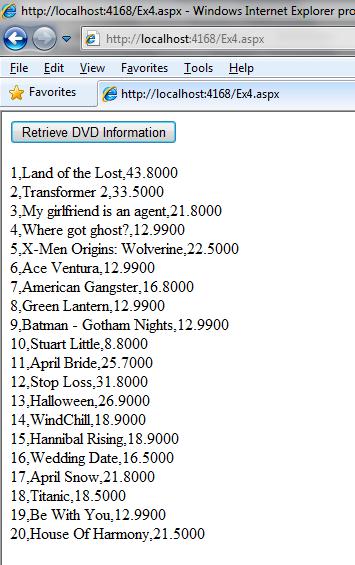


* 1. Build and run your application. Click on the button. What is displayed?



**Exercise 5:**

* 1. Continuing from your last exercise, add another button with ID=btnRetrieveProduct with Text = Retrieve Product.
  2. Write a program using Data Reader so that when the button is clicked, retrieve all records from the Product Table and display data from columns **ProductID**, **Title** and **Price** in the label.
  3. A sample output of the result is shown:



**========== End ==========**

