Code Girl T-SQL Problem Set 1

Environment Set Up

In order to learn T-SQL, you must have an environment to work in. You will need a running SQL Server instance and sample databases. SQL Server will only run on the Windows operating system. If you are running another OS, such as a MAC, you can contact me about accessing my SQL Server in the cloud. In that case, you will be responsible for setting up the Remote Desktop program on your MAC.

During Wednesday night Coder Girl meetups, I will bring media to install a free version of SQL Server on your Windows laptops.

If you would like to set up SQL Server yourself, watch these videos found on the Aunt Kathi Coder Girl page

How to get SQL Server cheap or free for learning

Install SQL Server Express

Set up AdventureWorks database

AdventureWorks Sample Database

Microsoft provides the AdventureWorks database for learning about databases and T-SQL. Many books and webinars use this database for the examples. At least to get started, we will also use AdventureWorks.

The AdventureWorks databases are available on CodePlex.

Schemas

One option in SQL Server is to set up schemas, kind of like containers for database tables. By default, if schemas are not used, the default schema is "dbo". You will often see databases that use the default schema for all tables.

When using schemas, you pretty much have to use both the schema and table name to refer to any table. Actually, this is a good practice even if the database has just dbo. Separate the schema and table name with a period. Here is an example:

Person.Address

"Person" is the schema, and "Address" is the table.

The AdventureWorks database takes advantage of schemas to group tables. In real life, schemas can also be used to manage permissions. For example, it's possible to give read-only permission to all the tables in a schema with one command instead of one command per table.

The schemas listed in AdventureWorks are HumanResources, Person, Production, Purchasing, and Sales.

Tables

There are many objects in a SQL Server database, but tables hold the data that we think about when we imagine a database.

There are several tables that I like to work with to demonstrate T-SQL. Here is a list of the ones I use the most:

Sales.SalesOrderHeader

Sales.Customer

Sales.SalesOrderDetail

Person.Person

Production.Product

Production.ProductCategory

Production.ProductSubCategory

HumanResources.Department

HumanResources.Employee

Sales.SalesPerson

Sales.SalesTerritory

Sales.SpecialOffer

Videos Required for Problem Set 1

The following videos are recommended before attempting the problems in this document:

Use Configuration Manager to stop or start SQL Server

How to use SQL Server Management Studio

Hello world and simple select

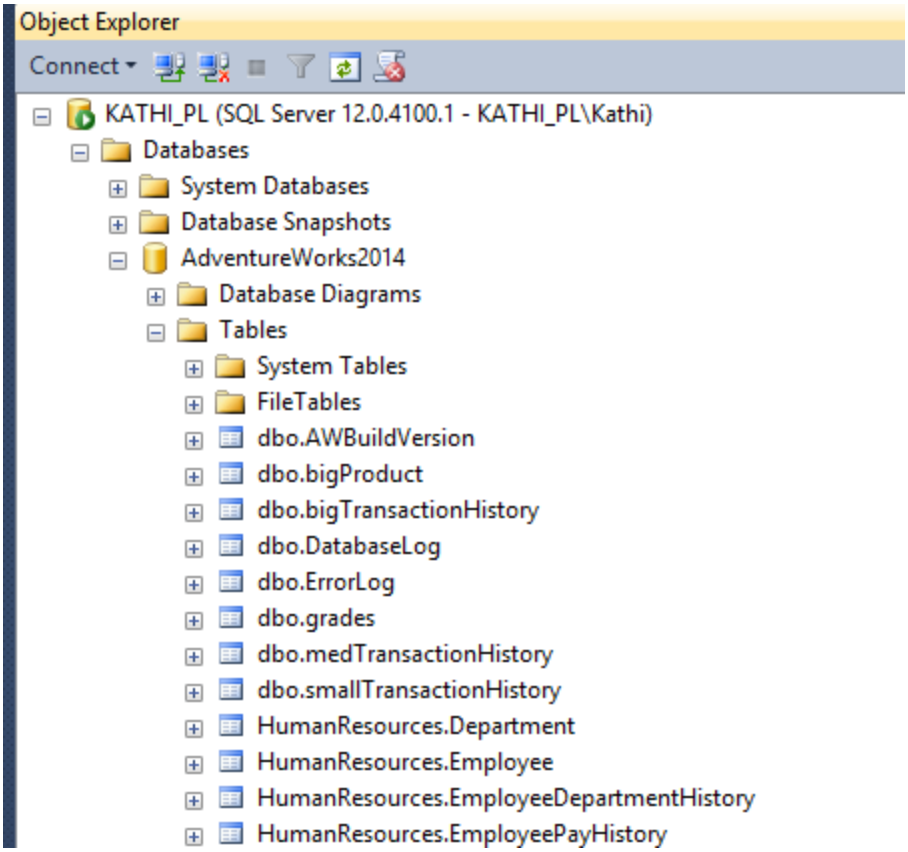
Why use square brackets?

Sort your results

Exercise 1

You can write some queries without typing a word. Use SQL Server Management Studio (SSMS) to solve the problems in this exercise.

1.1 Run SSMS and connect to your local SQL Server instance. In the Object Explorer window, expand until you can see the tables in the AdventureWorks2014 database.



1.2 Just using SSMS without typing, create and run queries that return up to 1000 rows from each of these tables:

Person.Person
Sales.Customer
Production.Product

Find the table. Right-click the table name and click “Select Top 1000 Rows.”

1.3 What are the two important keywords used in the queries you wrote in 1.2? One means a list of columns to be returned. One tells you which table the data comes from.

SELECT is the list of columns; FROM specifies the table.

1.4 Here is T-SQL query. The table name is fully qualified with three parts. What does each part refer to?

```
SELECT DeptartmentID, Name
FROM AdventureWorks2014.HumanResources.Department
```

AdventureWorks2014 is the database name; HumanResources is the schema name; Department is the table name.

Exercise 2

The tools in SSMS are great, but most of the time you will need to type out your T-SQL statement. To get a query window for typing your queries, you will need to click "New Query" in the menu. Make sure that the database listed in the dropdown box is the AdventureWorks2014 database.

2.1 What is the difference between these two statements?

```
PRINT 'Hello world!';

SELECT 'Hello world!';
```

PRINT displays the message in the Messages tab.

SELECT returns tabular data in a grid

2.2 What are two reasons for using brackets around column and table names?

The team that you work on requires brackets around all column and table names.

A table or column name is actually a keyword or has spaces.

2.3 Write a query that returns all of the rows and all of the columns from the Person.Person table. Use the * instead of typing out the columns.

```
SELECT * FROM Person.Person;
```

2.4 It is a bad practice to use the * instead of typing out the list of columns for work that you will turn in. Instead of using the *, write a query that returns the BusinessEntityID and the three name columns from Person.Person.

```
SELECT BusinessEntityID, FirstName, MiddleName, LastName
FROM Person.Person;
```

2.5 Write a query that returns a list of all the orders in the Sales.SalesOrderHeader table. Return the SalesOrderID, CustomerID, OrderDate, and TotalDue columns.

```
SELECT SalesOrderID, CustomerID, OrderDate, TotalDue
FROM Sales.SalesOrderHeader;
```

2.6. Starting with the SSMS shortcut you used in Exercise 1, write a query that returns the list of products (Production.Product table). Remove the columns except for ProductID, Name, Color, and ListPrice. Run your modified query.

```
/****** Script for SelectTopNRows command from SSMS  ******/
SELECT TOP 1000 [ProductID]
    ,[Name]
    ,[Color]
    ,[ListPrice]
FROM [AdventureWorks2014].[Production].[Product]
```

Exercise 3

When you run a query, the data will be returned in the order that is easiest for SQL Server unless you specifically sort it. Get more practice typing SELECT statements by writing these queries.

3.1 Write a query that returns the Name, ProductID, Color, Size, ListPrice, and ReorderPoint from the Production.Product table. Sort the query by Name.

```
SELECT Name, ProductID, Color, Size, ListPrice, ReorderPoint
FROM Production.Product
ORDER BY Name;
```

3.2 Starting with the query you wrote in 3.1, sort the results by ListPrice in descending order.

```
SELECT Name, ProductID, Color, Size, ListPrice, ReorderPoint
FROM Production.Product
ORDER BY ListPrice DESC;
```

3.3 Write a query that returns the SalesOrderId, OrderDate, TotalDue, and CustomerID from Sales.SalesOrderHeader. Sort by SalesOrderID.

```
SELECT SalesOrderID, OrderDate, TotalDue, CustomerID
FROM Sales.SalesOrderHeader
ORDER BY SalesOrderID;
```

3.4 Change the query you wrote in 3.3 so that the results are sorted by TotalDue in descending order.

```
SELECT SalesOrderID, OrderDate, TotalDue, CustomerID
FROM Sales.SalesOrderHeader
ORDER BY TotalDue DESC;
```

3.5 Write a query that returns information about the job candidates. See if you can find the table and schema name on your own. Return all of the columns in the table. Sort the results by BusinessEntityID.

```
SELECT JobCandidateID, BusinessEntityID, Resume, ModifiedDate
FROM HumanResources.JobCandidate;
```
