

Saumya (Sammy) Bhatnagar

SQL Server 2008 Portfolio

Email: Sammy.Bhatnagar@SetFocus.com

Phone: 201.491.9525

Introduction

Hello.

My name is Saumya (Sammy) Bhatnagar:.

This portfolio is designed to give examples of my development skills with SQL Server 2008.

These examples are taken from project assignments completed during the SQL Server 2008 Master's Program at SetFocus LLC. SetFocus is a certified Microsoft Gold partner and uses Microsoft certified courses throughout their program.

This program is a very intense 13 week course that covers everything about SQL Server 2008. This includes T-SQL code along with administrator responsibilities such as security, mirroring, replication, snapshots, isolation levels. Other services are included as well including SSIS, SSRS just to name a few.

I am confident that from this portfolio, you will be assured of my skills in SQL development.

Saumya (Sammy) Bhatnagar

Table Of Contents

Introduction	2
Piggy Bank Project	4
Data Diagram	5
Example of T-SQL Code.	6
Examples of Advance T-SQL Code	11
SSIS	15
SSRS	20
Recommendation.	23

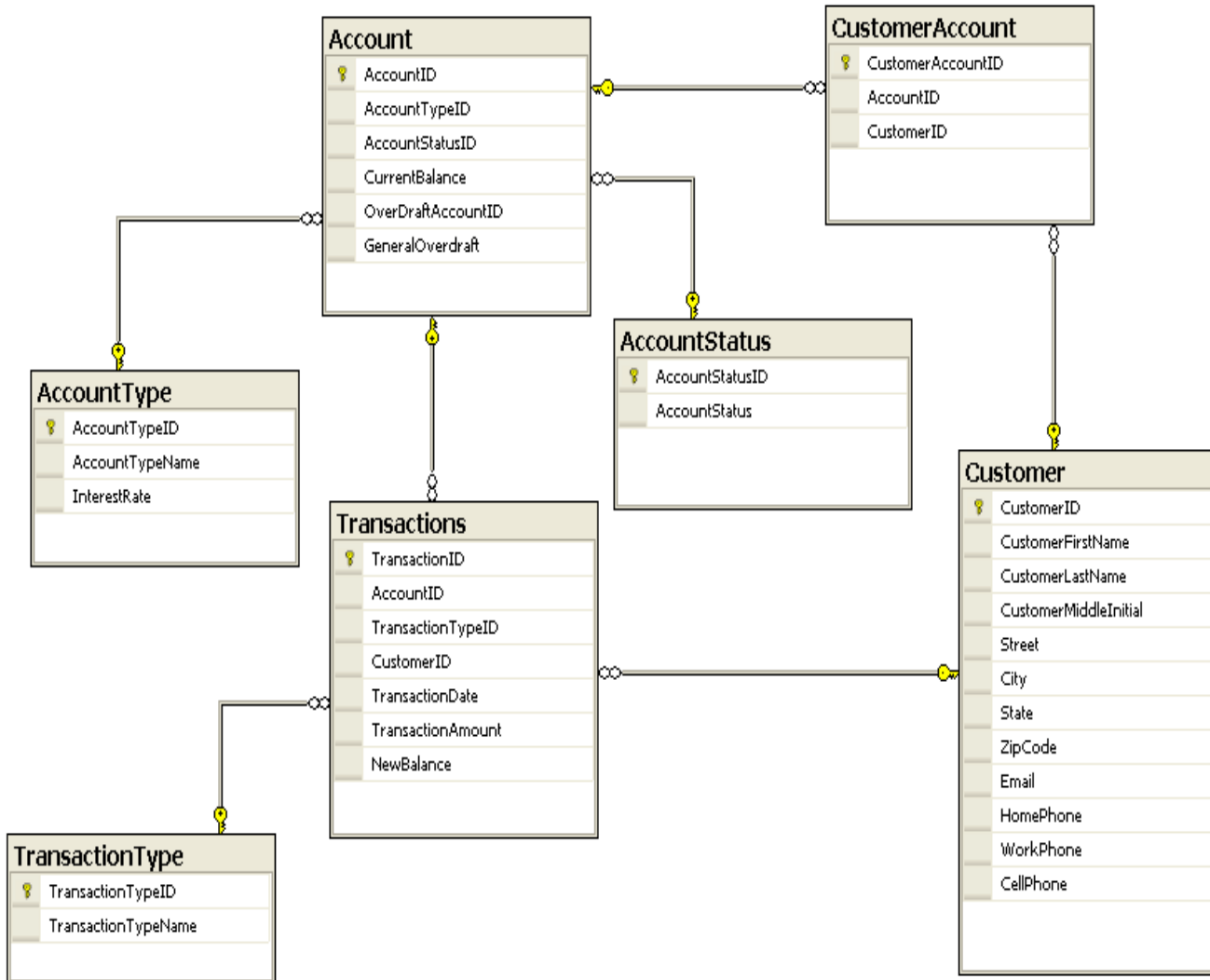
Piggy Bank Project

The Piggy Bank Project was an example of my T-SQL development skills. Piggy Bank is a bank which needed to track their Customers, Accounts and Transactions in one database.

In addition to Table and Column development, I was able to create stored procedures that were used to create customers, associate customers to accounts and perform transactions including deposits, withdrawals, overdrafts and transfers. In keeping with the data safety issues related to sensitive bank data, I also created Triggers that prevented deletion or updates to transactions and accounts.

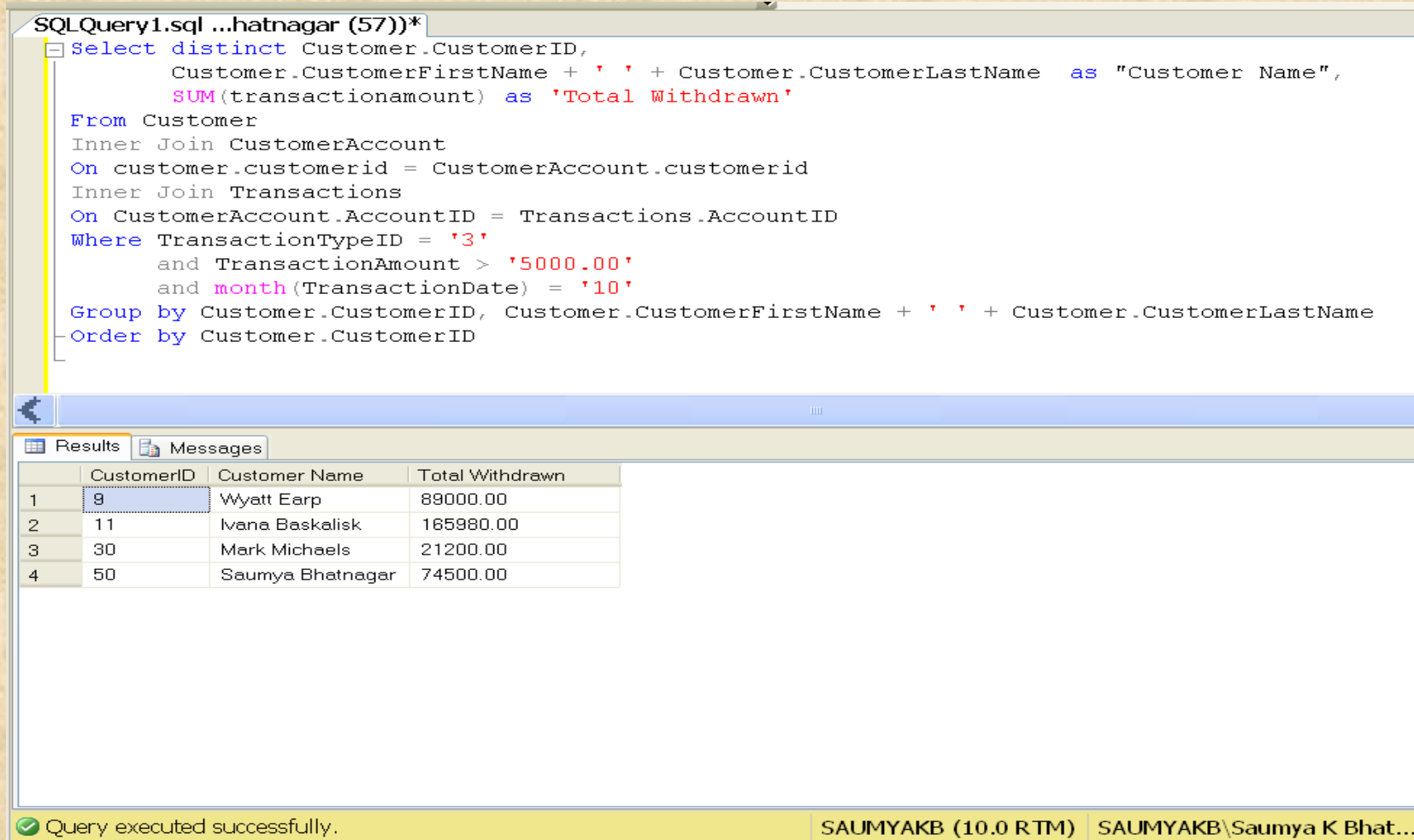
The database diagram follows. The following slides detail my SQL development skills.

Piggy Bank Project Database Diagram



Piggy Bank Project – Select Data

This is a T-SQL Code Sample that will list the Customer's ID, Name and Total Transaction Amount of those Customers who Withdrew more than \$5,000 at any point during the month of October. The report is sorted in descending order based on the Customer ID.



The screenshot displays a SQL Server Enterprise Manager window with a T-SQL query and its results. The query is as follows:

```
SQLQuery1.sql ...hatnagar (57))*  
Select distinct Customer.CustomerID,  
    Customer.CustomerFirstName + ' ' + Customer.CustomerLastName as "Customer Name",  
    SUM(transactionamount) as 'Total Withdrawn'  
From Customer  
Inner Join CustomerAccount  
On customer.customerid = CustomerAccount.customerid  
Inner Join Transactions  
On CustomerAccount.AccountID = Transactions.AccountID  
Where TransactionTypeID = '3'  
    and TransactionAmount > '5000.00'  
    and month(TransactionDate) = '10'  
Group by Customer.CustomerID, Customer.CustomerFirstName + ' ' + Customer.CustomerLastName  
Order by Customer.CustomerID
```

The results are displayed in a table with the following data:

	CustomerID	Customer Name	Total Withdrawn
1	9	Wyatt Earp	89000.00
2	11	Ivana Baskalisk	165980.00
3	30	Mark Michaels	21200.00
4	50	Saumya Bhatnagar	74500.00

At the bottom of the window, a status bar indicates: **Query executed successfully.** The user information is SAUMYAKB (10.0 RTM) SAUMYAKB\Saumya K Bhat...

Piggy Bank Project – Stored Procedure

This is an example of a Stored Procedure to 'Get Activity for a Certain Account' for the Current Month. Please do note the validation I put in to ensure that an Account ID is entered as a Parameter. Additionally, I created a view for the Transactions table in order to provide a level of security from the actual source tables.

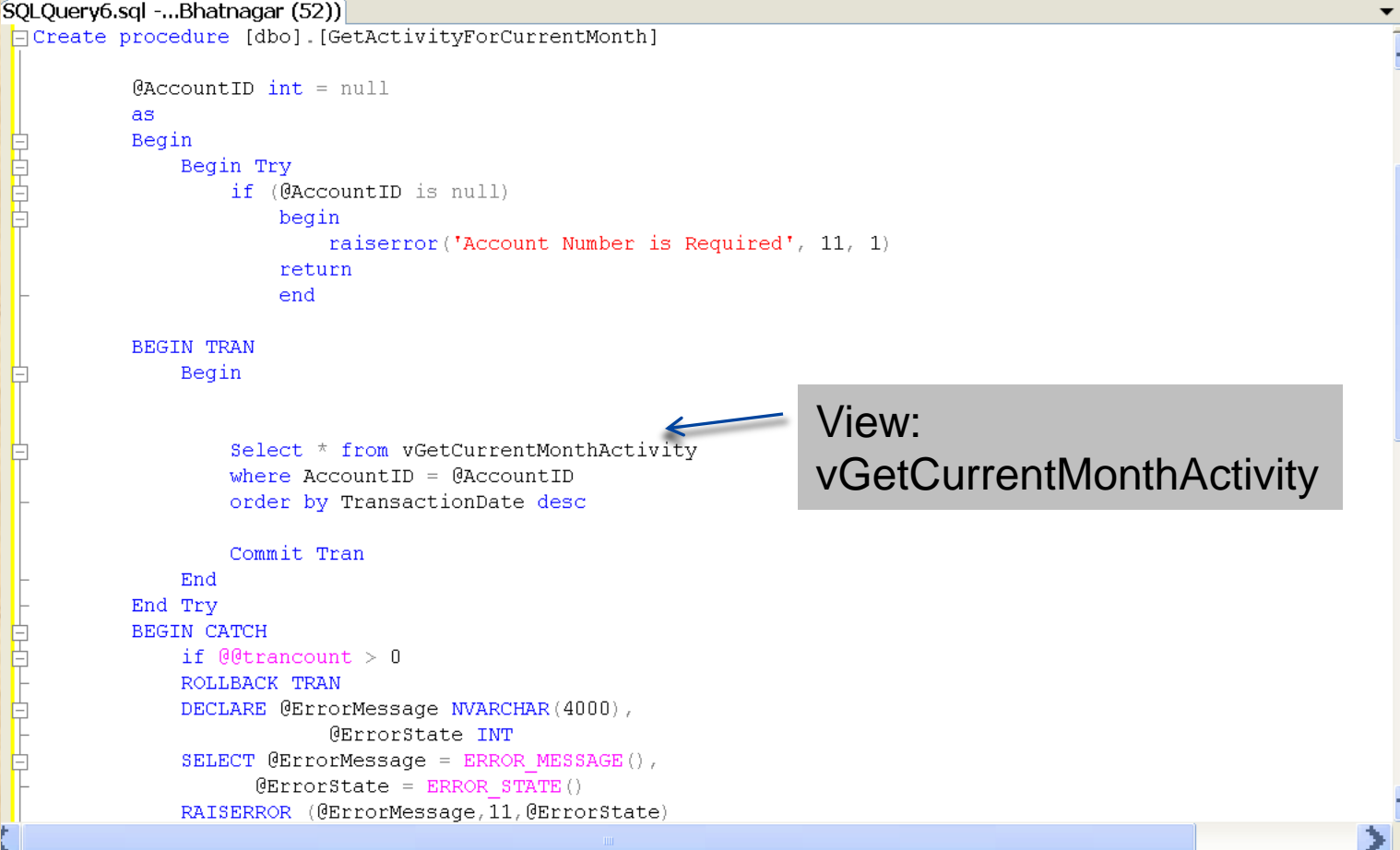
```
SQLQuery6.sql -...Bhatnagar (52))
Create procedure [dbo].[GetActivityForCurrentMonth]

    @AccountID int = null
as
Begin
    Begin Try
        if (@AccountID is null)
            begin
                raiserror('Account Number is Required', 11, 1)
                return
            end

        BEGIN TRAN
            Begin

                Select * from vGetCurrentMonthActivity
                where AccountID = @AccountID
                order by TransactionDate desc

                Commit Tran
            End
        End Try
    BEGIN CATCH
        if @@trancount > 0
            ROLLBACK TRAN
        DECLARE @ErrorMessage NVARCHAR(4000),
                @ErrorState INT
        SELECT @ErrorMessage = ERROR_MESSAGE(),
                @ErrorState = ERROR_STATE()
        RAISERROR (@ErrorMessage,11,@ErrorState)
```



View:
vGetCurrentMonthActivity

Connected. (1/1) SAUMYAKB (10.0 RTM) SAUMYAKB\Saumya K Bhat... PiggyBankProject 00:00:00 0 rows
Ln 18 Col 5 Ch 2 IP

Piggy Bank Project – Stored Procedure

This is an example of a Stored Procedure to Withdraw Funds from a specific account. Please do note the nested query which calls other stored procedures that will force a special or general overdraft charge to the account if there are insufficient funds. Note the validation script which alerts the user of withdrawals from an inactive account.

```
SQLQuery7.sql ...hatnagar (52))*  
  
if (@TransactionAmount > @CurrentBalance and @overdraftaccountid is not null)  
begin  
    exec ForceSpecialOverDraftAccountTransaction @accountid, @CustomerID, @TransactionAmount  
    return  
end  
if (@TransactionAmount > @CurrentBalance and @overdraftaccountid is null and @generaloverdraft = '1')  
begin  
    exec ForceGeneralOverDraftTransaction @accountid, @customerid, @transactionamount  
    return  
end  
if (@TransactionAmount > @CurrentBalance and @overdraftaccountid is null and @generaloverdraft is null)  
begin  
    raiserror('Insufficient Funds. Transaction Cancelled', 11,1)  
    return  
end  
  
IF exists(select * from Account  
    where Account.AccountID=@AccountID and  
    Account.AccountStatusID='2')  
BEGIN  
    Raiserror ('Withdrawals cannot be made from an Inactive Account', 11,1)  
RETURN  
END
```

Nested Stored Procedures within an "IF" Clause.

**Validation:
Withdrawals cannot be made from inactive accounts.**

Connected. (1/1) SAUMYAKB (10.0 RTM) SAUMYAKB\Saumya K Bhat... PiggyBankProject 00:00:00 0 rows

Piggy Bank Project – Trigger

This is an example of a Trigger which does not allow the Deletion/Alteration of any Table or View.

```
SQLQuery8.sql -...Bhatnagar (54))
CREATE TRIGGER [trigPBP]
ON database
for drop_table, alter_table, drop_view, alter_view AS
Begin
raiserror ('Tables and Views may not be Deleted or Altered',11,1)
ROLLBACK transaction
end
GO

SET ANSI_NULLS OFF
GO

SET QUOTED_IDENTIFIER OFF
GO

DISABLE TRIGGER [trigPBP] ON DATABASE
GO

ENABLE TRIGGER [trigPBP] ON DATABASE
GO
```

Tables may not be Deleted or altered

The screenshot shows the SQL Server Enterprise Manager interface. On the left, the Object Explorer displays the server hierarchy for 'SAUMYAKB (SQL Server 10.0.1600.1)'. The 'PiggyBankProject' database is expanded, showing 'Tables' and 'System Tables'. In the center, the SQL Query window shows the command 'drop table saumya'. Below the query window, the Messages pane displays the following error:

```
Msg 50000, Level 11, State 1, Procedure trigPBP, Line 5
Tables and Views may not be Deleted or Altered
Msg 3609, Level 16, State 2, Line 1
The transaction ended in the trigger. The batch has been aborted.
```

Piggy Bank Project – Trigger

This is another example of a Trigger which does not allow any Updates or Deletions to the Transactions Table.

```
SQLQuery9.sql -...Bhatnagar (54)
USE [PiggyBankProject]
GO

/***** Object: Trigger [dbo].[trigDontDeleteTransactions]    Script Date: 12/29/2009 17:09:02 *****/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

create trigger [dbo].[trigDontDeleteTransactions]
on [dbo].[Transactions]
instead of delete, update
as
Begin
    raiserror ('Transactions cannot be deleted or updated!', 11, 1)
    Rollback Tran
end
GO

Connected. (1/1) SAUMYAKB (10.0 RTM) SAUMYAKB\Saumya K Bha
```

Advance T-SQL Examples

Building upon the SQL skills that I have gained from the Piggy Bank Project Database and it's associated Stored Procedures, Views and Triggers, I would also like to demonstrate my more advanced T-SQL skills.

Various advanced T-SQL examples include:

1. Use of Common Table Expressions (CTEs)
2. PIVOT command
3. APPLY command

Advance T-SQL Examples

Below is an example of a **Common Table Expression (CTE)**. Common Table Expressions are temporary result sets similar to a derived table subquery.

```
Lab1-CTE.SQL -...hatnagar (54)* Lab2-PIVOT.SQL...hatnagar (51)*
;WITH BooksCTE AS
(
    SELECT BookTitlePK,
           BookISBN,
           BookCategory,
           BookName
    FROM DimBooks
    WHERE BookCategory IN ('SQL Server', 'Oracle')
) ,

Orders2007CTE AS
(
    SELECT BookTitlePK,
           SUM(SalesDollars) as TotSalesDollars
    FROM FactBookSales
    JOIN DimBookPrice on DimBookPrice.BookPricePK = FactBookSales.BookPricePK
    JOIN DimDates ON FactBookSales.DatePK = DimDates.DatePK
    WHERE DimDates.Year = 2007
    GROUP BY BookTitlePK)

select BooksCTE.BookCategory, BooksCTE.BookName, BookISBN, TotSalesDollars
       from BooksCTE
       join Orders2007CTE
       on BooksCTE.BookTitlePK = Orders2007CTE.BookTitlePK
       order by BookCategory, TotSalesDollars DESC
```

Advance T-SQL Examples

Below is an example of a **Pivot Command** using a CTE. The Pivot Command allows users to transform rows of data into analytical columns. Successful Pivot commands require ability to write aggregate functions. I was able to use the pivot commands to create reports in SQL Server Reporting Services.

Lab2-PIVOT.SQL...hatnagar (51))*

```
WITH SalesCTE AS
(
    SELECT CustomerCategory,
           SUBSTRING(QuarterKey, 6, 1) AS QuarterNumber,
           SUM(SalesDollars) AS TotDollars
    FROM FactBookSales
    JOIN DimDates on FactBookSales.DatePK = DimDates.DatePK
    join DimCustomers on FactBookSales.CustomerPK = DimCustomers.CustomerPK
    WHERE Year = 2006
    GROUP BY CustomerCategory, SUBSTRING(QuarterKey, 6, 1))

SELECT CustomerCategory, [1] as Q1, [2] as Q2, [3] as Q3, [4] as Q4
    from SalesCTE
        PIVOT( SUM(TotDollars) FOR QuarterNumber in ( [1], [2],[3], [4])) as TempList
```

Result Set from
the Pivot Command

	CustomerCategory	Q1	Q2	Q3	Q4
1	College Book Store	1336725.00	1209187.00	1299134.00	1237380.00
2	Distribution	1944409.00	1979139.00	2054922.00	1928244.00
3	Internet	4692679.00	4680814.00	4721137.00	4916202.00
4	Retail Store	2776068.00	2669839.00	2649886.00	2718218.00

Advance T-SQL Examples

Below is an example of an **Apply function**. This function allows the user to apply the results of a Function to each row of a result set.

Lab3-APPLY.SQ...atnagar (53))*

```
CREATE FUNCTION [dbo].[GetTopNCustomersSalesDays]
( @CustomerPK int, @NumberDays int, @StartDate datetime, @EndDate datetime )
RETURNS TABLE

AS

RETURN
    SELECT TOP ( @NumberDays) WITH TIES ActualDate, SUM(SalesDollars) as TotDollars
    FROM FactBookSales
        JOIN DimDates ON FactBookSales.DatePK = DimDates.DatePK
        WHERE CustomerPK = @CustomerPK
            AND ActualDate BETWEEN @Startdate and @EndDate

    GROUP BY ActualDate
    ORDER BY TotDollars DESC

GO

declare @StartDate Datetime, @EndDate datetime
set @StartDate = '1/1/2007'
SET @EndDate = '2007-2-28'

SELECT CustomerName, ActualDate, TotDollars
FROM DimCustomers
cross apply dbo.GetTopNCustomersSalesDays (DimCustomers.CustomerPK, 5, @StartDate , @EndDate )
```

SQL Server Integrated Services (SSIS)

The SSIS Package Developed Was To Import Validated Data Into A Smaller Version Of The AdventureWorks Database. This New Database Is Named MiniAdventureWorks.

Several Data Source Files Were Provided. The Data In These Source Files Are As Follows:

- Shipping Methods. Various shipping methods/companies available.
- Products. The Current Listing of Products Available Along With Its Relevant Information.
- Vendors. The Current Listing of Vendors Along With Their Relevant Information.
- Purchase Orders. This Information Is The Most Complex. The Purchase Order Will Only Be Added To The Database If All Of Its Information Is Correct.
The Vendor And Product(s) Associated With the Purchase Order MUST Exist.

The Purchase Order Information Is Split Into Two Separate Tables.

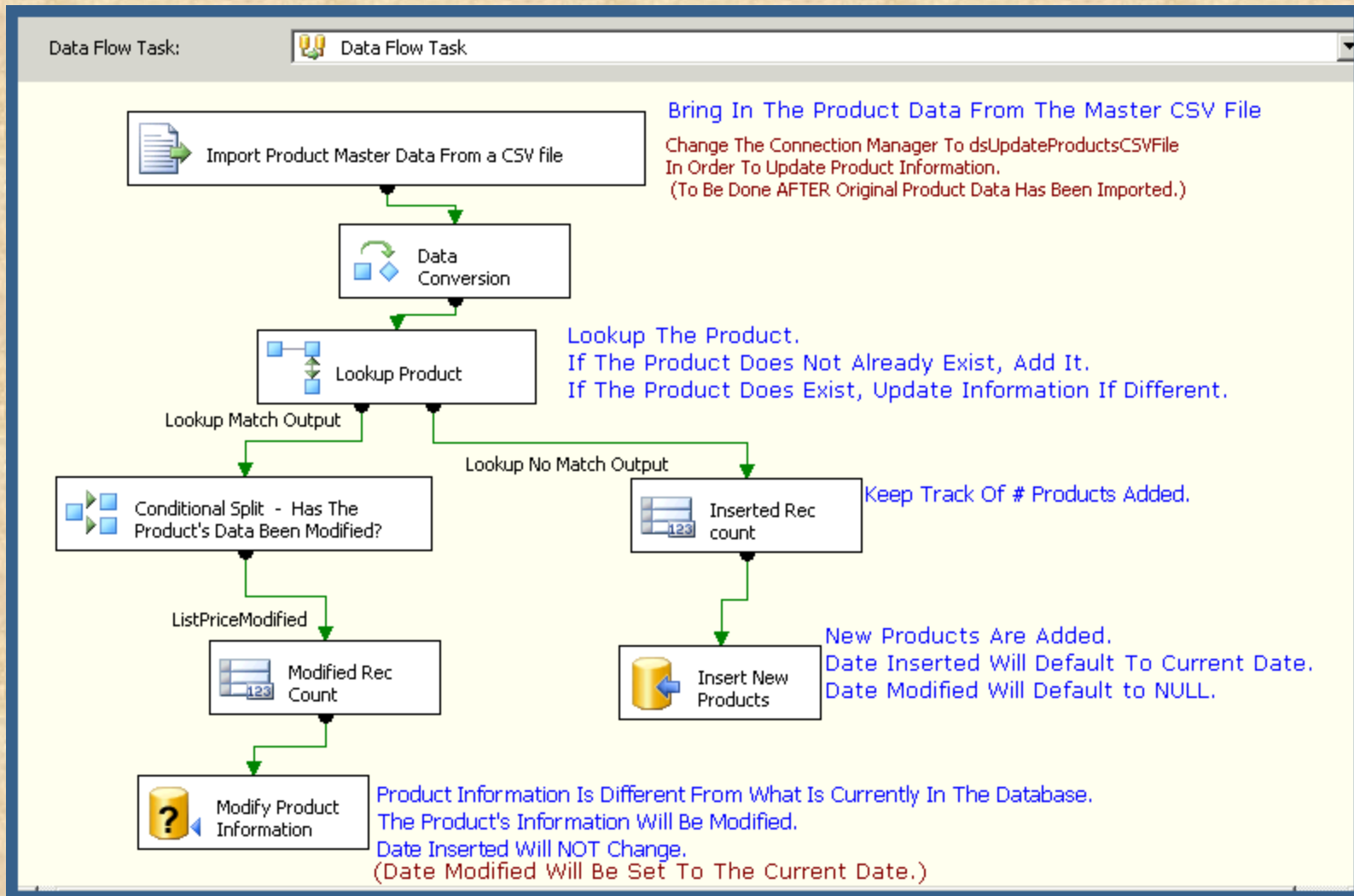
The First Table Is The Purchase Order Header. This Table Contains Information About The Purchase Order As A Whole... Total Price, Date Order Was Placed, Vendor Placing The Order, And Shipping Information, Just To Name A Few.

The Second Table Is The Purchase Order Detail. This Table Contains The Information About Each Product That Was Ordered... Quantity Ordered, Price Per Unit, Total Price.

Updates To Product Data And Addition Of New Products were taken into consideration. For example, When A Product Is Encountered For A Second Time (Update Source File), The Information Is Checked Against The Current Information In The Database. If The Data Is Different, The Information Will Be Updated.

SQL Server Integrated Services (SSIS)

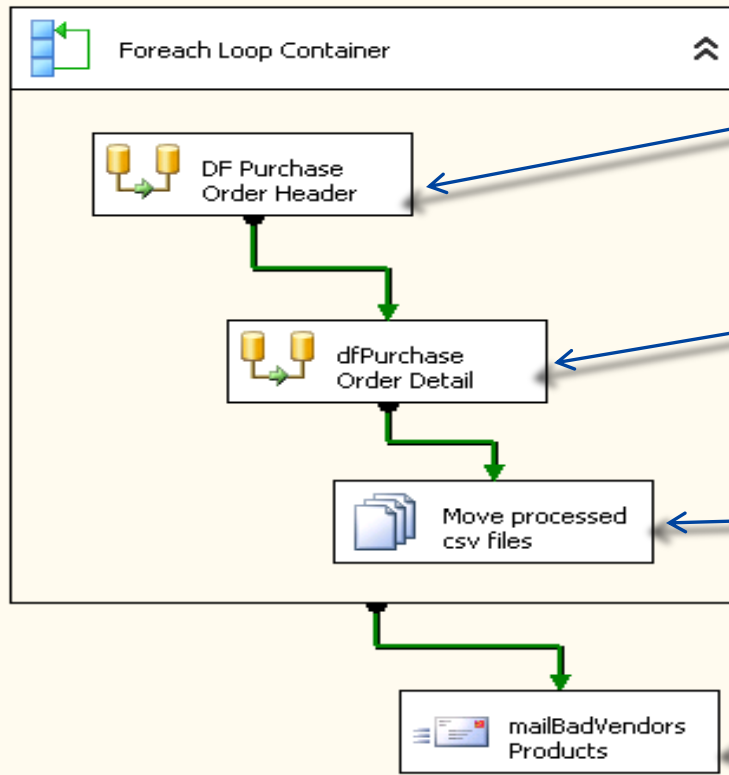
Data Flow Task For The Product Import



SQL Server Integrated Services (SSIS)

Control Flow Example For The Purchase Order Data

This SSIS package will import PO Data from .a source file into the MiniAdventureWorksDB Database



Process PO Header Data

Process PO Detail Data linking the PO Header Data

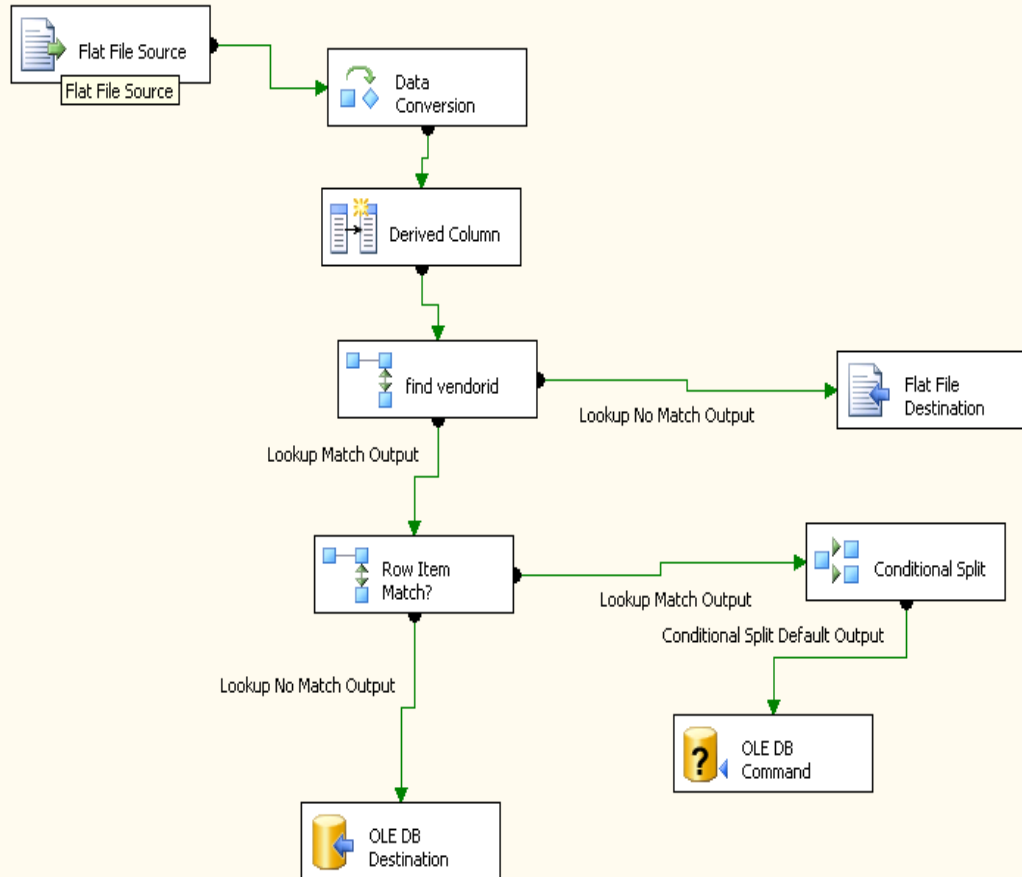
Move Source Files to a predetermined location

Send Mail Task to email .csv file with invalid vendor names to a specified email address

SQL Server Integrated Services (SSIS)

Data Flow Example For The Purchase Order Data

This SSIS package will import PO Data from .a source file into the MiniAdventureWorksDB Database



Invalid Vendor Data Is Written Out To A Text File (.csv) for review at a later date.

SQL Server Reporting Services (SSRS)

The following are examples of SSRS Reports that were created throughout the course of the Master's Program. Not only do the following demonstrate my abilities to create reports in SSRS, but they also demonstrate my abilities to use the Advanced T-SQL skills as well.

The following Report Examples are:

1. **"Vendor Sales by Year"** – This report utilizes a parameter definition where the user chooses a specific 'Ship Method' by which to sort the data. In this report, I used my newly developed skills in using CTEs and the Pivot Command.
2. **"Most Active Customers for a Given Month"** – This report details the most active users of a company's product and displays, in descending order, what those user's activity was for the given month of October. In this report, I was able to utilize my newly developed skills in Aggregate SQL Functions as well as using the 'Rank' command to sort the data from highest to lowest.
3. **"Unfiled Reports by Date"** – This report is an example of surveys that have not been filled out by a given date. In this report, I demonstrate how I can develop a report that utilizes multiple input parameters to display data for multiple Survey Dates.

SQL Server Reporting Services (SSRS)

Vendor Sales By Year – For Ship Method 'Truck Ground'

Please Enter ShipMethod: XRQ - TRUCK GROUND

12/30/2009 10:58:53 AM **Vendor Sales by Year using Ship Method:** SFDOMAIN\sammy.bhatnagar
XRQ - TRUCK GROUND

Vendor Name	Year 2002	Year 2003	Year 2004	Total
Chicago Rent-All	\$2,208	\$7,174	\$18,764	\$28,146
Circuit Cycles	\$27,951	\$80,042	\$181,682	\$289,674
Compete Inc.	\$15,246	\$91,476	\$210,037	\$316,759
Compete Enterprises Inc	\$111,937	\$447,747	\$1,007,431	\$1,567,116
Custom Frames Inc.	\$20,957	\$73,724	\$184,107	\$278,788
Federal Sport	\$5,105	\$16,592	\$43,393	\$65,090
First National Sport Co.	\$8,016	\$26,051	\$68,133	\$102,200
Gardner Touring Cycles	\$1,819	\$5,913	\$15,465	\$23,198
International Sport Assoc.	\$11,140	\$51,987	\$126,253	\$189,380
Morgan Bike Accessories	\$1,257	\$5,446	\$14,244	\$20,948
SUPERSALES INC.	\$1,575	\$6,300	\$18,375	\$26,250
Victory Bikes	\$6,994	\$20,790	\$52,166	\$79,949
Total	\$214,204	\$833,242	\$1,940,051	

Utilized CTE and PIVOT Command to arrange data By Year.

SQL Server Reporting Services (SSRS)

Most Active Customers with Orders for the Month of November

SFDOMAIN\sammy.bhatnagar

12/30/2009 11:00:43 AM

Most Active Customers for

November

Customer	Number of Orders
John B Doe	4
Michele A Assare	3
Paul C Raber	3
Joseph A Morissey	2
Bill C Joe	2
Tommy M Giustino	2
Georges L Nieves	1
Melanie G Jerome	1
Paul G Stockwell	1
Shannon I Rosa	1

Utilized Aggregate SQL function as well as Rank Function to display data from highest to lowest

SQL Server Reporting Services (SSRS)

Incomplete Course Evaluations by Date

12/30/2009 11:02:26
AM

SFDOMAIN\sammy.bhatnagar

Unfilled Evaluations By Date

as of: 11/5/2009,11/15/2009,11/16/2009,11/20/2009,11/21/2009,11/22/2009,12/3/2009,12/9/2009,12/14/2009

Evaluation ID	Create Date
4	11/5/2009 12:00:00 AM
14	11/15/2009 12:00:00 AM
15	11/16/2009 12:00:00 AM
19	11/20/2009 12:00:00 AM
20	11/21/2009 12:00:00 AM
21	11/22/2009 12:00:00 AM
32	12/3/2009 12:00:00 AM
38	12/9/2009 12:00:00 AM
50	12/14/2009 12:00:00 AM
56	12/14/2009 12:00:00 AM

Example of Multiple
Input Parameters for
Report By Date

Letter of Recommendations

SetFocus

Bridging the Gap between Talent and Opportunity.SM

4 Century Drive
Parsippany, NJ
07054

Tel: 973.889.0211
Fax: 973.889.0212

www.setfocus.com

December 16, 2009

To Whom It May Concern:

It is my pleasure to provide a recommendation for Sammy Bhatnagar. Graduating from the SetFocus SQL Server 2005 Master's Program, Sammy Bhatnagar excelled in an intense, integrated, in-depth, full time, 3 month SQL Server Administrator's Training Program.

I was impressed by Sammy's ability to work with many technologies during his time with SetFocus. Sammy has gained experience with the latest Microsoft SQL Server 2005 Technologies, including but not limited to designing, implementing and maintaining a SQL Server Database, writing queries with Transact SQL, troubleshooting and optimizing database servers using Microsoft SQL Server 2005, report writing with SSRS, Designing a BI Solution using SQL Server 2005, and XML.

I found Sammy's work ethic was demonstrated by his ability to work in an environment with firm deadlines, very large workloads, and complex specifications. Sammy creates code in a time efficient manner, is flexible with specifications changes and is able to work within teams on assigned projects.

I recommend Sammy as a solid addition to your staff. If you would like more information, I'd be happy to provide it.

Sincerely,



Robert Witkowski
Dean of Students
rwitkowski@setfocus.com

