**Create Views**

CREATE VIEW Client\_Addresses AS

SELECT ClientID, ClientFirstName, ClientLastName,

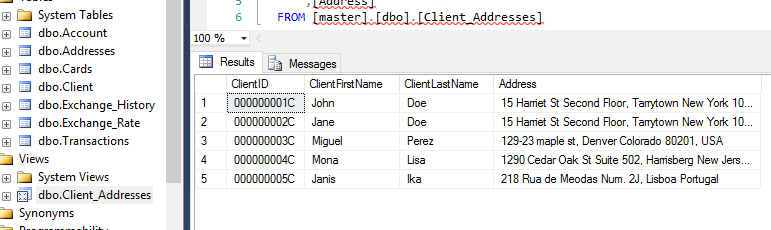
(Street1 + IIF((Street2 IS NOT NULL), ' ' + Street2, '')

+ ', ' + City + ' ' +

IIF((State IS NOT NULL), (State + ' ' + CONVERT(varchar, Zipcode) + ', '), '')

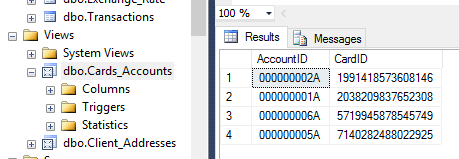
+ Country) AS Address

FROM Client INNER JOIN ADDRESSES ON ClientAddress=AddressID



CREATE VIEW Cards\_Accounts AS

SELECT Account.AccountID, Cards.CardID FROM Account INNER JOIN Cards ON Cards.AccountID=Account.AccountID;



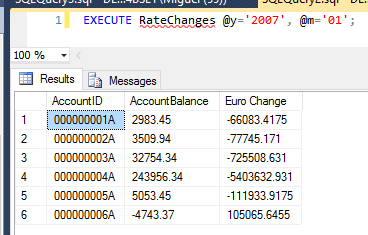
**Stored Procedures**

CREATE PROCEDURE RateChanges @y AS smallint, @m AS tinyint AS

SELECT AccountID, AccountBalance, (AccountBalance\*EUR\_Change) AS 'Euro Change'

FROM Account, Exchange\_History WHERE MONTH(Percent\_Month\_ID)=@m AND YEAR(Percent\_Month\_ID)=@y

ORDER BY AccountID ASC;



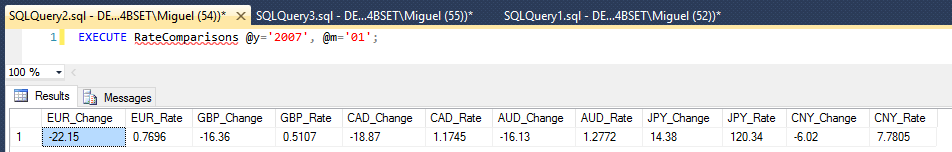
CREATE PROCEDURE RateComparisons @y AS smallint, @m AS tinyint

AS

SELECT EUR\_Change, EUR\_Rate, GBP\_Change, GBP\_Rate, CAD\_Change, CAD\_Rate, AUD\_Change, AUD\_Rate, JPY\_Change, JPY\_Rate, CNY\_Change, CNY\_Rate

FROM Exchange\_Rate, Exchange\_History WHERE MONTH(Percent\_Month\_ID)=@m AND YEAR(Percent\_Month\_ID)=@y AND Percent\_Month\_ID=Rate\_Month\_ID

ORDER BY Percent\_Month\_ID DESC;



**Triggers**

CREATE TRIGGER DuplicateInExchangeHistory ON Exchange\_Rate

AFTER INSERT

AS

DECLARE @MonthID AS date = (SELECT Rate\_Month\_ID FROM inserted)

DECLARE @EUR AS float = ((((SELECT EUR\_Rate FROM inserted)-0.9886)/0.9886)\*100)

DECLARE @GBP AS float = ((((SELECT GBP\_Rate FROM inserted)-0.6106)/0.6106)\*100)

DECLARE @CAD AS float = ((((SELECT CAD\_Rate FROM inserted)-1.4477)/1.4477)\*100)

DECLARE @AUD AS float = ((((SELECT AUD\_Rate FROM inserted)-1.5227)/1.5227)\*100)

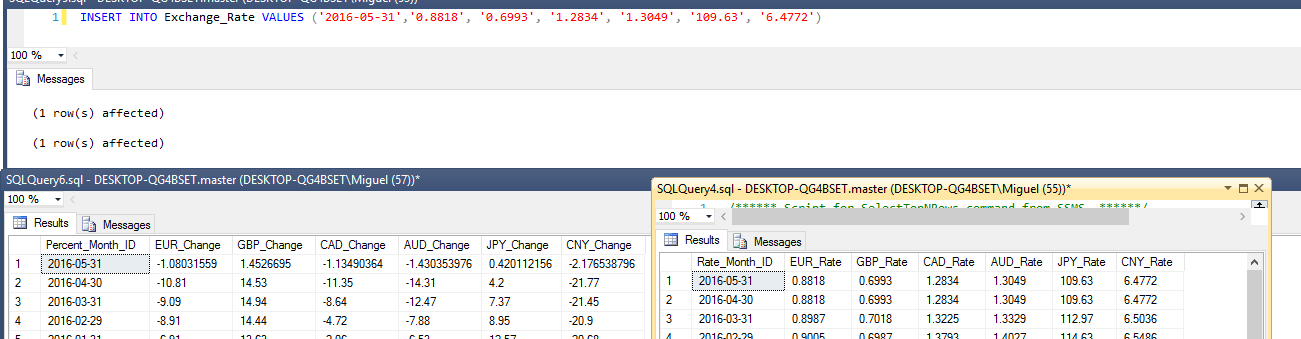
DECLARE @JPY AS float = ((((SELECT JPY\_Rate FROM inserted)-105.21)/105.21)\*100)

DECLARE @CNY AS float = ((((SELECT CNY\_Rate FROM inserted)-8.2792)/8.2792)\*100)

INSERT INTO Exchange\_History VALUES (@MonthID, @EUR, @GBP, @CAD, @AUD, @JPY, @CNY)

GO

-- I realized I had a math error much later on because it was off by a factor of ten, I switched from \*10 to \*100



CREATE TRIGGER UpdateCNYRate ON Exchange\_History

FOR UPDATE

AS

UPDATE Exchange\_Rate

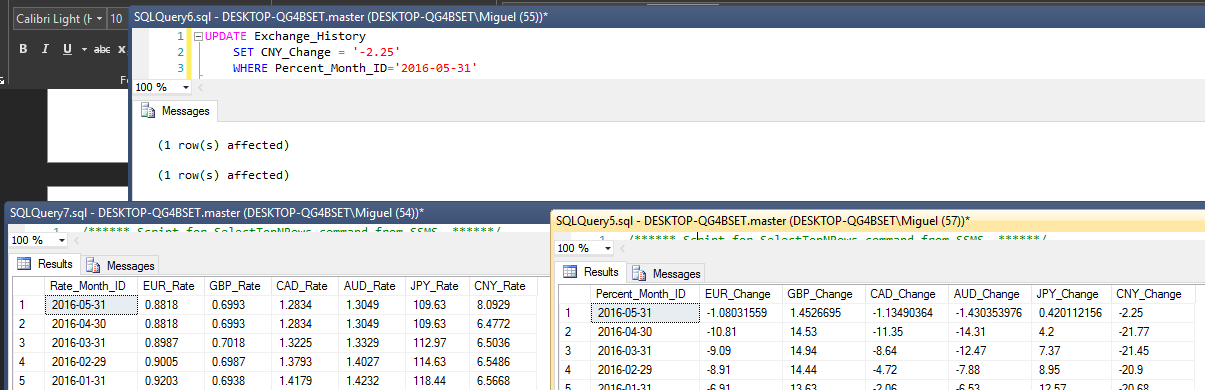
SET CNY\_Rate = (SELECT (CNY\_Rate) FROM Exchange\_Rate WHERE Rate\_Month\_ID='2000-01-31') \*

(1+((SELECT (Exchange\_History.CNY\_Change) FROM Exchange\_History, inserted

WHERE Exchange\_History.Percent\_Month\_ID = inserted.Percent\_Month\_ID)/100))

FROM inserted WHERE Exchange\_Rate.Rate\_Month\_ID = Percent\_Month\_ID

GO



**Using External XML Script**

DECLARE @x XML

SELECT @x=P FROM OPENROWSET(BULK '~filepath~', SINGLE\_BLOB) AS Products(P)

DECLARE @hdoc int

EXEC sp\_xml\_preparedocument @hdoc OUTPUT, @x

UPDATE Exchange\_Rate

SET EUR\_Rate = (SELECT EUR FROM OPENXML (@hdoc, '/root/Exchange\_Rate', 2) WITH (EUR money)),

GBP\_Rate = (SELECT GBP FROM OPENXML (@hdoc, '/root/Exchange\_Rate', 2) WITH (GBP money)),

CAD\_Rate = (SELECT CAD FROM OPENXML (@hdoc, '/root/Exchange\_Rate', 2) WITH (CAD money)),

AUD\_Rate = (SELECT AUD FROM OPENXML (@hdoc, '/root/Exchange\_Rate', 2) WITH (AUD money)),

JPY\_Rate = (SELECT JPY FROM OPENXML (@hdoc, '/root/Exchange\_Rate', 2) WITH (JPY money)),

CNY\_Rate = (SELECT CNY FROM OPENXML (@hdoc, '/root/Exchange\_Rate', 2) WITH (CNY money))

WHERE Rate\_Month\_ID = (SELECT Month\_ID FROM OPENXML (@hdoc, '/root/Exchange\_Rate', 2) WITH (Month\_ID date))

EXEC sp\_xml\_removedocument @hdoc

-- XML FILE (final.xml)

<root>

<Exchange\_Rate>

<Month\_ID>2016-05-31</Month\_ID>

<EUR>0.8819</EUR>

<GBP>0.6994</GBP>

<CAD>1.2835</CAD>

<AUD>1.3050</AUD>

<JPY>109.64</JPY>

<CNY>8.0930</CNY>

</Exchange\_Rate>

</root>

