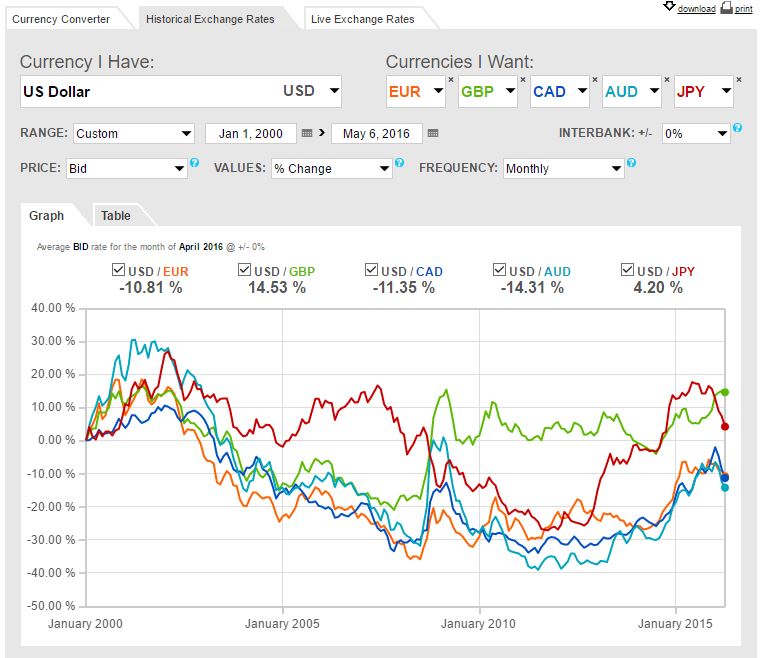
**Incorporating External Data**

1. <https://www.oanda.com/currency/historical-rates/> Historical Exchange Rates
   1. USD to EUR GBP CAD AUD JPY / 1-1-00 to 5-6-16 / Bid / %Change / Monthly
   2. USD to EUR GBP CAD AUD JPY / 1-1-00 to 5-6-16 / Bid / Rate / Monthly
2. Contains 196 Data points for each currency conversion, for both datasets
3. I chose this dataset as a recommendation by the professor. This particular data source has business services that offers real time access to changing rates, so I thought it would be a reliable source for historical data. I chose both percent change and actual rates to be able to generate a greater variety of data. The fluctuation of currency and how it would affect the value of an account over time is interesting. On a higher level, one can see how well an economic region is doing compared to others based on the valuation of the country’s currency. If I could add a 6th I would add the Chinese Yuan. When it first started appearing in late 1993 it rose in valuation tremendously, but has been slowly declining ever since, until recently within the last 2 years. I know I don’t technically need the percentage rates, however it made for a good graph to include, and even though I could derive the percentages mathematically, I felt it would have been less system intensive to already have the values instead of calculating them real time, every time.



**Joining Your Tables**

I used the import feature in MSSQL Server. I had some difficulty getting it to work, however I finally managed after installing some additional office plugins. The original file I got was in CSV format, however since everything was already laid out in a table format, I just modified some fields, removed others, saved it to XSLX to then make the import process simpler. I actually went through a few import scenarios in order to better understand how the tool actually works. One thing I disliked was the inability to make more granular changes with the wizard without resorting to code, like adding a primary key for example.

|  |  |  |  |
| --- | --- | --- | --- |
| **Attributes** | **Description** | **Entity** | **Data Type** |
| **Rate\_Month\_ID** | **The month and year associated with that rate (PK)** | **Exchange\_Rate** | **date** |
| EUR\_Rate | The value of 1 US Dollar in Euros | Exchange\_Rate | smallmoney |
| GBP\_Rate | The value of 1 US Dollar in British Pounds | Exchange\_Rate | smallmoney |
| CAD\_Rate | The value of 1 US Dollar in Canadian Dollars | Exchange\_Rate | smallmoney |
| AUD\_Rate | The value of 1 US Dollar in Australian Dollars | Exchange\_Rate | smallmoney |
| JPY\_Rate | The value of 1 US Dollar in Japanese Yen | Exchange\_Rate | smallmoney |

**Updating your Logical Model**

Even though I am not using Exchange\_History for this, I am still going to include it in the ERD diagram and MSSQL database for later use. I decided to make the values calculated based on the information in Exchange\_Rate to derive Exchange\_Chart using an INNER JOIN

|  |  |  |  |
| --- | --- | --- | --- |
| **Exchange\_Chart** | | | |
| **Attributes** | **Description** | **Entity** | **Data Type** |
| AccountID | The unique identifier for an account | Exchange\_Rate | char(10) |
| Rate\_Month\_ID | The month and year associated with that rate | Exchange\_Rate | date |
| AccountBalance\_EUR | Account Balance in Euros | Exchange\_Rate | smallmoney |
| AccountBalance\_GBP | Account Balance in British Pounds | Exchange\_Rate | smallmoney |
| AccountBalance\_CAD | Account Balance in Canadian Dollars | Exchange\_Rate | smallmoney |
| AccountBalance\_AUD | Account Balance in Australian Dollars | Exchange\_Rate | smallmoney |
| AccountBalance\_JPY | Account Balance in Japanese Yen | Exchange\_Rate | smallmoney |



**Test Data**

I used the task import data to load the source files into the database. I had to make some modifications so as to create a Primary Key during the import since that was not an option.

|  |  |  |
| --- | --- | --- |
|  |  |  |

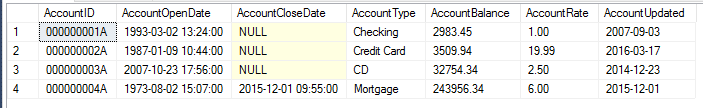
ALTER TABLE Account ADD AccountUpdated date;

UPDATE Account SET AccountUpdated='09/03/2007' WHERE AccountID='000000001A';

UPDATE Account SET AccountUpdated='03/17/2016' WHERE AccountID='000000002A';

UPDATE Account SET AccountUpdated='12/23/2014' WHERE AccountID='000000003A';

UPDATE Account SET AccountUpdated=CAST(AccountCloseDate AS date) WHERE AccountID='000000004A';



SELECT Account.AccountID, Exchange\_Rate.Rate\_Month\_ID,

Account.AccountBalance,

(Account.AccountBalance\*Exchange\_Rate.EUR\_Rate) AS AccountBalance\_EUR,

(Account.AccountBalance\*Exchange\_Rate.GBP\_Rate) AS AccountBalance\_GBP,

(Account.AccountBalance\*Exchange\_Rate.CAD\_Rate) AS AccountBalance\_CAD,

(Account.AccountBalance\*Exchange\_Rate.AUD\_Rate) AS AccountBalance\_AUD,

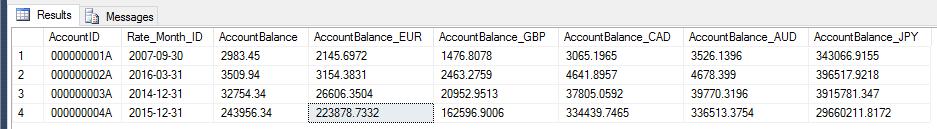
(Account.AccountBalance\*Exchange\_Rate.JPY\_Rate) AS AccountBalance\_JPY

FROM Exchange\_Rate INNER JOIN Account

ON (YEAR(Account.AccountUpdated)=YEAR(Exchange\_Rate.Rate\_Month\_ID)

AND MONTH(Account.AccountUpdated)=MONTH(Exchange\_Rate.Rate\_Month\_ID))

ORDER BY AccountID ASC;



|  |  |
| --- | --- |
|  | Added an excel spreadsheet just to check the math to make sure everything was calculated correctly. |