4 HW #4: Joins

Answer the following questions using only the syntax discussed in class. If a year is unspecified, please use the 2010 data and refer to the data dictionary for questions regarding the contents of the data.

4.1 Linking Stocks to Fundamental Data

We are looking to create a D2010-centric join to fnd. For each row in D2010 we want to join it to the appropriate row in fnd. An important note is that the retdate in fnd represents the date that information was released. For each row in D2010 we want to join it to the most recent row, backward looking, in fnd of the appropriate company.

To make this process clear, we focus on a single stock: Apple.

1. Using the fnd data, locate Apple computers and determine their 2010 revenue, net income and cash.
2. Locate Apple's 2010 10K, which is their annual, SEC mandated, financial report. Match the numbers above to the financial report.[4] Please put a screenshot on these numbers, from Apple's 10K, in your homework to verify that you found them.
3. Write a query which returns the apple data from d2010. How many rows are there?
4. Write a query which returns the appropriate (Apple) information from the link table (lnk). Only consider those links (rows) which have linktype equal to "LU" or "LC" and have linkprim equal to "P" or "C." The rest of the links are considered too weak to be useful.
5. For a join to be correct it must match the appropriate identifier (gvkey, permno and permco) and also be within a valid time frame. The columns "linkdt" and "linkenddt," in lnk, refer to when a link becomes valid and is no longer valid (inclusive), respectively. If linkenddt is equal to "E" that means that it is currently valid. Write a query which joins D2010, as the left-hand-side, to lnk. Take care handle the dates properly, even though in the case of Apple there is no need. Feel free to select all columns from all tables for this query.
6. Verify that no rows were created or destroyed by this join.
7. If you join the above to fnd using **only** gvkey, how many rows are in the resulting dataset? Why?

[4] For cash, you should look for the "Consolidated Balance Sheet," while the other two numbers can be found in the "Consolidated Statements of Operations."

8. To make sure that no rows are created we will restrict our join to rows that are within a year of retdate in fnd. In particular, the join condition should make sure that retdate from D2010 is between fnd's retdate and a year from that date. For this query feel free to select all the columns from all three tables. Verify that the result of your query returns the appropriate number of rows.

Now that we understand how to do this for the data from Apple, we are going to repeat the process for the entire 2010 table which will require dealing with some additional minor issues.

1. Begin by filling out the following table:

Table Name	Number of Rows
lnk	
fnd	
d2010	
d2011	

2. How does the number in the above change if we only consider strong links?

3. On the stock side, let's look at d2010, which is keyed on permno and verify that it, yields the same results: no nulls. Note that permno is a security level identifier, while permco is a *company* level identifier. Since a company can have multiple stocks, there can be multiple permno numbers associated with a single permco. Identify the permco's that have multiple rows in d2010. How many of them are there?

4. A unique company in fnd is gvkey, which is a company level identifier. Verify that there are no rows with missing gvkeys. Since we are also matching on retdate, also verify that there are no null retdates.

5. However, we do have a problem. Describe what the following query returns and why it could be problematic if we join as we had joined the Apple data.

   ```
   select count(1)
           , gvkey
           , date_part('year', retdate::date)
           from fnd
           group by 2,3
           having count(1) > 1
   ```

6. Because of the above, we won't be able to simply look forward 1 year to determine the range of dates which fnd matches d2010. To merge the data we will use a second

link table which will contain the following:

- gvkey
- retdate
- the next retdate, chronologically

Create this second link table by doing the following:

(a) select gvkey, retdate as the left hand side table

(b) select gvkey, retdate as *nextret* as the right hand side

(c) join on gvkey and have all lhs retdates less than (strictly) nextret

(d) In the outer query select the min(nextret), which we will call nextret, grouping by gvkey and retdate.

Gvkey '180599' was one of the problematic rows. Show the data from this gvkey after the procedure above is completed. What does this query do? We will call this procedure the gvkey-date-crossjoin.

7. Now that we understand the structure of the data we can join. Starting with D2010, left join the lnk table, left join the gvkey-date-crossjoin table and then left join against fnd, returning only the following columns:

   - Identifers: gvkey, permco and permno
   - fnd data: cash, emp, netinc, rev, retdate as fnddt, tic and company name
   - d2010 data: prc, ret, vol and shrout

   Make sure to verify that the number of rows in your table is appropriate. To facilitate this process, I recommend using a table which you can delete and repopulate. The syntax for this is:

```
create table stocks2016.jointmp as (SELECT ...)
```

   and to delete the table use:

```
drop table stocks2016.jointmp;
```

4.2 Other Questions

1. Answer the following using the d2010 data:

(a) We define the equal-weighted returns as the average return, per day, for all stocks. Write a query which returns the equal-weighted returns for each day in 2010.

(b) We define the market-weighted returns as the average return, per day, weighted by the total dollar value of shares traded for each stock. Write a query which returns the daily market-weighted return for 2010.

(c) What is the market-weighted return on January 21st?

(d) Write a query that returns the market weighted, equal weighted return and the return date, sorted by date. Plot them.

(e) Which stocks had more than 150 days where their daily return was greater than the average equal-weighted return over the entire year?

(f) Which stock had the biggest absolute difference between its max and min daily return

(g) Write a query that returns the date where the market return and equal weighted return are most different.

2. For this last question, you may have to run multiple queries:

   - What happened more frequently: Stocks in 2010 outperforming the average daily-return of all stocks in 2011 or stocks in 2011 outperforming the average daily-return of all stocks in 2010?

3. The percentage of trading days in 2010 that are trading days in 2011.[5]

4. The 5 stocks which had the biggest reversals from their return in 2010 to 2011 (The stocks that had the biggest change in their *yearly* return. Yearly return is defined as the price on the last day that a stock is trade subtracting the price on the first day it is traded divided by the stock price on the first day. Pay particular attention to potential missing data issues.

5. The number of stocks alphabetically before each stock (e.g. A would be 0, AA would be 1, etc.) in 2010.[6]

6. Looking at the data per-month, the number of stocks that have their best month ((price at end of month - price on first day of month)/(price on first day of month)) in January of 2010. The price should be from the end and beginning of the month, respectively. Make sure to handle the case if a stocks only appears intermittently

[5] As in class, consider a trading day to be the same when the day and month match.

[6] Comparisons of the form *string* $\leq$ *string* do alphabetical comparison. Also keep in mind that you can join using any conditional expression.

within a month. In that case, you should consider the first time a stock appears during a month and the last time it appears as the open and close dates.

7. What is the probability that a stock which increase in price on month will do in the next month?[7]

8. How many missing days are there? Make sure to only count missing days **after** a stock has been in the data. So if a stock doesn't appear in the data until February, January does not count as missing. If a stock leaves the market before the end of the year, you can either count the days past their exit as missing or as not missing, just be consistent across all stocks.[8]

9. In finance we often create a measure called "abnormal returns," which looks at the returns of a stock after removing some market factor effects. One of those effects is the "market itself" in the form of "β", the perceived correlation between the market's risk and the stock's risk. You can think about this along the lines of "How related is this stock to the overall market?" As you can imagine, some stock's movements mirror the overall market, such as casinos, while others only tangentially relate to how the market moves, such as firms whose primary line of business is government contracts. We are going to look at some "β"s here.

   (a) Calculate the market's daily return, either equal or market-weighted, making sure to ignore those days with zero total volume, write a single query for each of 2010 and 2011.

   (b) Calculate the correlation (using any function you choose) between each stock's daily return and the market's return. Do this for both 2010 and 2011, writing a single query for each.[9]

   (c) Using whatever software tool you choose, create a scatter plot which shows β against that stock's yearly return for all stocks in 2010 and 2011. There should be two points for each stock, one for each year, they should be differentiated by color or shape.

   (d) Do you notice any pattern in the above?

[7] Be wary of null values in December!

[8] In other words, if a stock leaves the data it maybe because the stock delisted, in which case the data is not missing.

[9] There is a correlation function in PostgreSQL, feel free to use it.