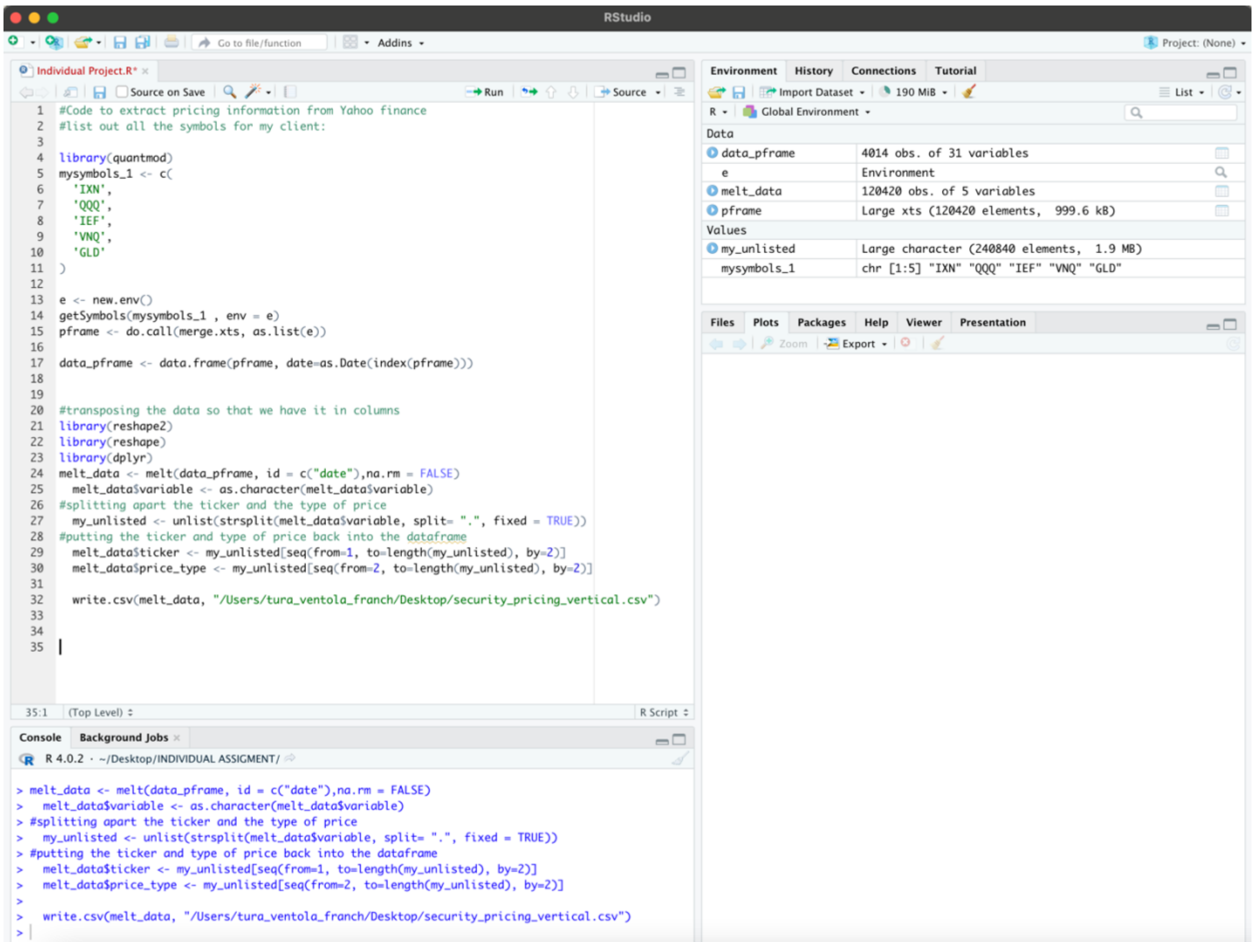
STEP 1: Downloading the data

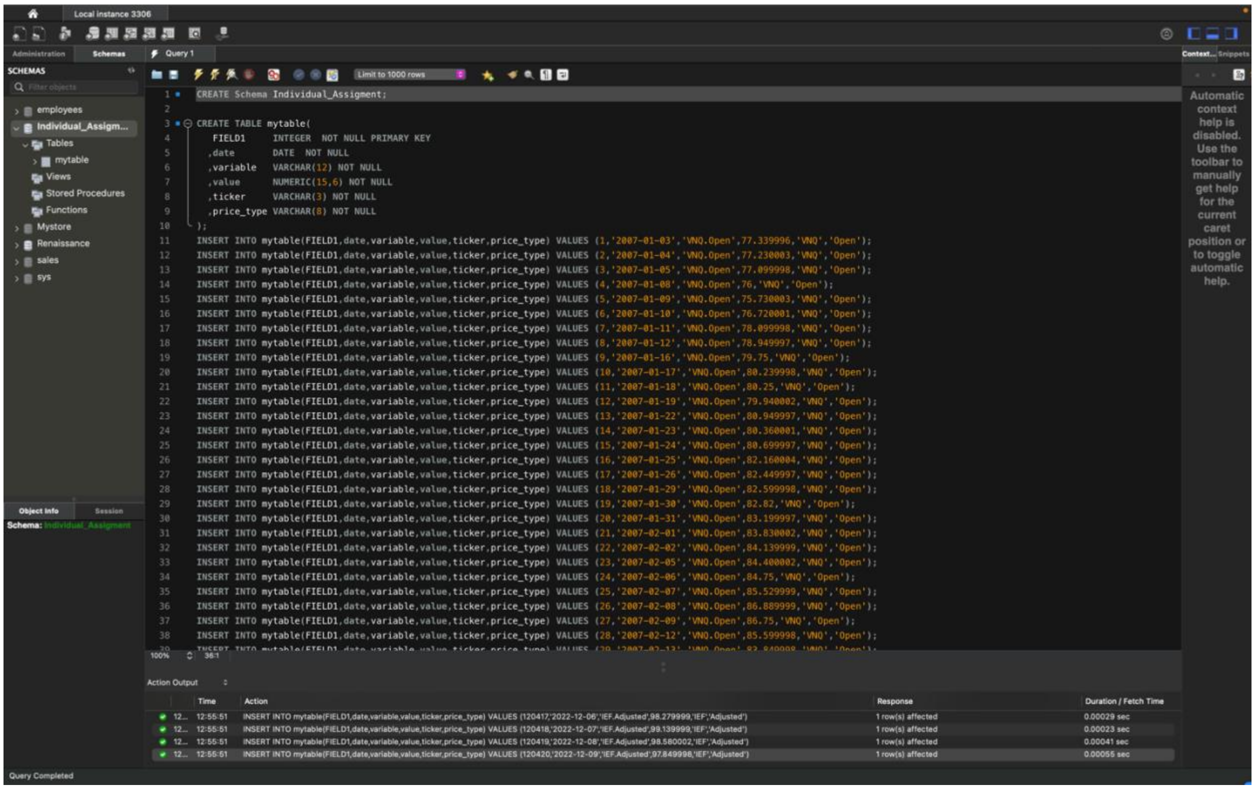
Using the library "quantmod" in R, we are able to extract the daily prices from Yahoo Finance about the tickers our client has in his portfolio.



With this code we obtain a csv file with all the pricing data.

STEP 2: Loading the data

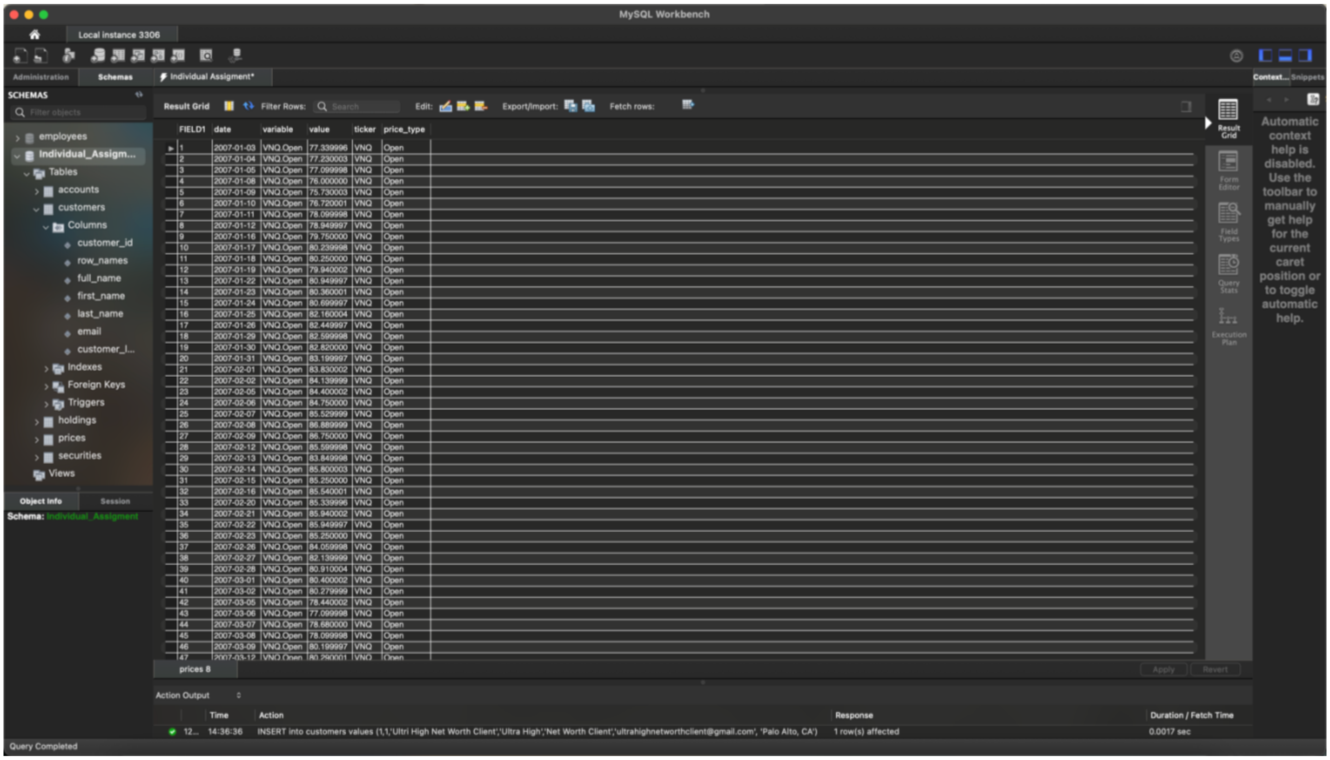
We use the following page: "https://www.convertcsv.com/csv-to-sql.htm" to convert the csv file obtained in step 1 to an SQL code that will create a table with the data.



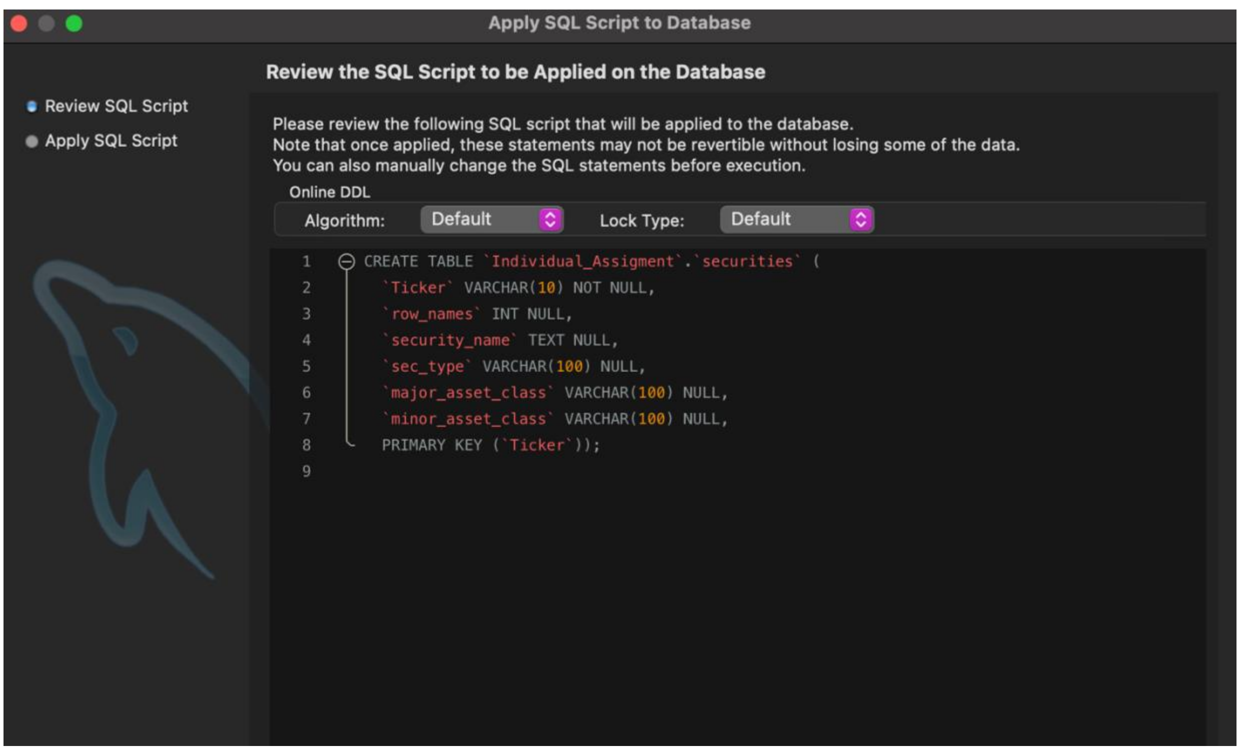
First we need to create a schema to have a structure for our database. Once its created, we paste the code from the website and we obtain our first table.

We renamed the table as prices for convenience.

Below we can see the results of our table:



We created the other tables needed with the data provided for our investor.



<u>STEP 3:</u>

3.1) Returns

First, we need to confirm the last date available for data coincides with the date the data was downloaded.

```sql
#Find max date
SELECT MAX(date)
from prices;
```

MAX(date)
2022-12-09

To find the returns for each of the securities, we are going to use the continuous return. This can be calculated by doing the natural logarithm (ln) of the current value (P1) divided by the initial price (P0).

In order to get the initial value in our case denominated as lagged price, we use the lag value. Since we're trying to obtain the returns for the most recent 12, 18 and 24 months, we use the number of market days in a month (approximately 21 days) and multiply it by the number of months we want the value for. Hence, we obtain 250 market days in 12 months, 375 for 18 months and 500 for 24 months. We put this numbers in the lag function, that looks back the number of days described before, and accesses the data of that row from the current one.
In the lag function we include partition by ticker and we order by date to indicate when to reset the computation.

```sql
#Q1
SELECT a.date, a.ticker,
        LN(a.value / a.price_12m) AS continuous_return_12M,
        LN(a.value / a.price_18m) AS continuous_return_18M,
        LN(a.value / a.price_24m) AS continuous_return_24M
FROM
(SELECT *,
    LAG(value, 250)
    OVER(PARTITION BY ticker
    ORDER BY date)
        AS price_12M,
    LAG(value, 375)
    OVER(PARTITION BY ticker
    ORDER BY date)
        AS price_18M,
    LAG(value, 500)
    OVER(PARTITION BY ticker
    ORDER BY date)
        AS price_24M
   FROM prices
   WHERE price_type = "Adjusted") a
   WHERE date = '2022-12-09';
```

Below we can see the continuous results for each ticker divided into three columns for the three periods.

date	ticker	continuous_return_12M	continuous_return_18M	continuous_return_24M
2022-12-09	GLD	0.00035920493925841216	-0.02395361303205936	-0.0403574063038234
2022-12-09	IEF	-0.15169992088880946	-0.13530110708982013	-0.17822754980757352
2022-12-09	IXN	-0.2887366945406856	-0.1322661024330806	-0.012971960014385994
2022-12-09	QQQ	-0.32364815513247364	-0.1813676148724042	-0.07417474413871572
2022-12-09	VNQ	-0.23241215050821176	-0.14373763333679157	0.06804483732183311

As we can see, most of the returns are negative, which means the investments are not profitable. In the past year only gold (GLD) has been profitable and QQQ was the worst, closely followed by IEF that are both categorized as Equity as an Asset Class. This makes sense from a macroeconomics perspective since in a situation of insecurity, equity tends to go down and the safest investments are commodities.

Moreover, year and a half ago no securities had positive returns, and two years ago only Vanguard (VNQ).

In order to take a look at the return of the portfolio as a whole, we need to take into account the weight of each asset in the portfolio. To do so, we sum the ln(P1/P0) multiplied by % of each asset (named quantity in our table) and we use a left join (in this case an inner join would produce the same outcome), to join the table holdings which contains the quantity. We unite both tables by ticker.

```
#Q1
#Portfolio's Return
SELECT a.date,
        SUM(LN(a.value / a.price_12M)*(b.quantity)) as portfolio_return_12M,
        SUM(LN(a.value / a.price_18M)*(b.quantity)) as portfolio_return_18M,
        SUM(LN(a.value / a.price_24M)*(b.quantity)) as portfolio_return_24M
FROM
(SELECT *,
    LAG(value, 250)
    OVER(PARTITION BY ticker
    ORDER BY date)
        AS price_12M,
    LAG(value, 375)
    OVER(PARTITION BY ticker
    ORDER BY date)
        AS price_18M,
    LAG(value, 500)
    OVER(PARTITION BY ticker
    ORDER BY date)
        AS price_24M
   FROM prices
   WHERE price_type = "Adjusted") a
   left join holdings as b
   on a.ticker=b.ticker
   WHERE date = '2022-12-09';
```

If we look at the returns of the portfolio as a whole, we can see that the results are not only negative, but have been decreasing even more from 2 years ago until now.

date	portfolio_return_12M	portfolio_return_18M	portfolio_return_24M
2022-12-09	-0.18589170554140877	-0.12009160669753728	-0.07268377608056842

3.2) Correlation

The most current MySQL environment does not allow to compute the correlation. Hence, we are going to compare the variances from each asset.

We decided to use a period of 6 months since it seemed neither too short not to have enough data for the calculation, nor too long to be influenced by macroeconomic events.

To calculate the variance, we just have to use the VARIANCE code with the continuous returns computed as before but lagged over the new period length. In order to compute the variance, we need more than one date to properly measure the fluctuation, if we had it like before (='2022-12-09'), we wouldn't be able to see any difference since we are only considering one day.

```
#Q2
SELECT z.ticker,
VARIANCE(z.continuous_returns) AS variance
FROM
(SELECT a.*, (a.value - a.lagged_price)/a.lagged_price as discrete_returns,
LN(a.value/a.lagged_price)as continuous_returns
FROM
(SELECT *, LAG(value, 125) OVER(
                                PARTITION BY ticker
                                ORDER BY date)
                                AS lagged_price
FROM prices
WHERE price_type = 'Adjusted'
AND date >= '2022-06-09') a
) as z
GROUP BY z.ticker;
```

ticker	variance
GLD	0.00015465325717295514
IEF	0.000038263865647377793
IXN	0.001416349665533533
QQQ	0.0013191562087886965
VNQ	0.0010582174037093595

Not surprisingly, the lowest return corresponds to IEF, which is Fixed Income. F.I. is the known for giving steady returns and low risk. On the other hand, equity is completely the opposite, returns tend to be more spread compared to the mean, which is why IXN and QQQ have the highest variance.

Overall, we can speculate that the variances are not significantly high but need more information to answer it properly.

3.3) Risk

To calculate the risk you can simply use the STD() function on the continuous return calculated as the previous sections (we wrote the risk as sigma). Since they ask us for the most recent 12 months, we only need to lag for 250.

In investing standard deviation is used to measure the security's volatility. Generally, we want a lower sigma since a higher one means that there's more volatility thus higher risk, however this depends on the risk aversion of the investor.

Risk move in line with the variation. The lowest risk corresponds to the Fixed Income security since as explained before it characterizes with steady returns and low risk, while equity has the highest risk since it also generates higher returns when the market is prosperous.

If the risk of a security seems pretty high, this can be complemented by less aggressive securities in the same portfolio. Thus we cannot make a proper conclusion of the volatility of out portfolio without looking at the risk as a whole.

```sql
#Q3
SELECT z.ticker,
    AVG(z.continuous_return) AS mu,
    STD(z.continuous_return) AS sigma
FROM
(SELECT a.*,
        LN(a.value / a.lagged_price) AS continuous_return
FROM
(SELECT *,
    LAG(value, 250)
    OVER(PARTITION BY ticker
    ORDER BY date)
        AS lagged_price
   FROM prices
   WHERE price_type = "Adjusted") a
   WHERE a.date >= '2021-12-09') z
   GROUP BY z.ticker;
```

ticker	mu	sigma
GLD	-0.005674923964952649	0.06204730296444215
IEF	-0.10073517532761872	0.05154652514539344
IXN	-0.06947022436259903	0.17837489208544322
QQQ	-0.08963771385168648	0.18271758496207646
VNQ	0.00610628987535463	0.1881938091807592

Calculating the risk of the portfolio is not as simple as before. The standard deviation of a portfolio is a function of how each holding in the portfolio moves in relation to the other holdings in the portfolio. Mostly because if the securities are correlated, when one performs badly, its highly likely than the other one will as well.

Hence, in order to calculate it we need the relationship between the securities. The formula to calculate the portfolio's risk is as following:

$$\sigma_p = \sqrt{{w_1}^2{\sigma_1}^2 + {w_2}^2{\sigma_2}^2 + 2w_1w_2Cov_{1,2}}$$

Where:

σ is the standard deviation

w is the weight of a security in the portfolio

$Cov_{1,2}$ is the covariance between two assets

Since as correlation, MySQL does not allow to compute the Covariance, for simplicity and to at least have an idea of the general risk, we calculate the risk of the portfolio as before, multiplying sigma (the risk) by the quantity (weight of each asset) and adding it up.

```sql
#Portfolio
SELECT SUM(sigma*(b.quantity)) as sigma_portfolio
FROM (
SELECT z.ticker,
    AVG(z.continuous_return) AS mu,
    STD(z.continuous_return) AS sigma,
    (AVG(z.continuous_return) / STD(z.continuous_return)) AS risk_adj_return
FROM
(SELECT a.*,
        LN(a.value / a.lagged_price) AS continuous_return
FROM
(SELECT *,
    LAG(value, 250)
    OVER(PARTITION BY ticker
    ORDER BY date)
        AS lagged_price
   FROM prices
   WHERE price_type = "Adjusted") a
   WHERE a.date >= '2021-12-09') z
   GROUP BY z.ticker) s
   left join holdings as b
   on s.ticker=b.ticker;
```

sigma_portfolio
0.11730708075691787

Even though this doesn't represent the real portfolio risk, it is a good indicator of it to compare it with other portfolios or to see the change if we were to modify (buy or sell) our current portfolio.

3.4) Recommendation

As seen in section 3.1, the returns of this portfolio are negative, which means our client is losing money and has been for the past two years. Since the negative returns have lingered over 2 years, we would recommend a drastic restructure or the portfolio.

Assuming he wants to maintain a similar structure regarding the asset classes of the portfolio, we did a bit of research on the highest performing securities for each asset class.
With that research we repeated the first two steps and step 3.1 to see if these new securities were indeed profitable. However, in the first tries, we still got a lot of negative returns in at least one period. *(*can see some of the references at the end)*

After numerous tries, we finally got securities with positive returns for all asset classes except for Real Assets. The specific security our investor currently has, is a REIT (real estate investment trust). However, when putting the top performing REITs on the SQL and calculating the result, we couldn't find any that has a positive return and out of those, VNQ, which is the one our investor currently has in his portfolio, had the highest performance.
Since we our basing our new portfolio in a similar structure as the current one, we decided to leave that security but with a lower weight since according to research the REITs market is not projected to improve in the immediate time.

Finally, although GLD had a positive return, we decided to take a look to other Commodities to see if there was a better option. We found out numerous alternatives with higher returns than gold which is why we decide to sell it as well.

Overall we recommended to sell all holdings except for VNQ (with a lower percentage). As for our buying recommendations, as said previously, we tried to maintain a similar structure as before, giving more weight to commodities since they have the highest returns and increasing diversity in equity.

<u>New recommendations:</u>

Equity: (34.6% compared to previous 39.6%)
First Investors Growth & Income A (FGINX) 10.7%
Federated Strategic Value Dividend A (SVAAX) 8.9%
Hennessy Cornerstone Value Inv (HFCVX) 15%

Fixed income: (22% compared to previous 28.5%)
SPDR Bloomberg 1-3 Month T-Bill ETF (BIL)

Real assets: (maintain) (2.4% compared to previous 8.9%)
Vanguard Real Estate ETF (VNQ)

Commodities: (41% compared to previous 23%)

Abrdn Bloomberg All Commodity ETF (BCD) 15%

United States 12-Month Natural Gas ETF (UNL) 26%

3.5) Change in risk and returns

We placed the data from the securities from 3.4 into a new table (*mytable*) and created a new table with the % each ticker represents (from *holdings_new*).

To see the changes in the portfolio we repeated the same procedures as sections 3.1 and 3.3 with the new tables.

We checked the data for the tickers separately first to ensure all returns are positive (excluding VNQ that we already knew was negative).

```
#Q5
SELECT a.date, a.ticker,
        LN(a.value / a.price_12m) AS new_return_12M,
        LN(a.value / a.price_18m) AS new_return_18M,
        LN(a.value / a.price_24m) AS new_return_24M
FROM
(SELECT *,
    LAG(value, 250)
    OVER(PARTITION BY ticker
    ORDER BY date)
        AS price_12M,
    LAG(value, 375)
    OVER(PARTITION BY ticker
    ORDER BY date)
        AS price_18M,
    LAG(value, 500)
    OVER(PARTITION BY ticker
    ORDER BY date)
        AS price_24M
   FROM mytable
   WHERE price_type = "Adjusted") a
   WHERE date = '2022-12-09';
```

```sql
SELECT z.ticker,
    AVG(z.continuous_return) AS new_mu,
    STD(z.continuous_return) AS new_sigma
FROM
(SELECT a.*,
        LN(a.value / a.lagged_price) AS continuous_return
FROM
(SELECT *,
    LAG(value, 250)
    OVER(PARTITION BY ticker
    ORDER BY date)
        AS lagged_price
   FROM mytable
   WHERE price_type = "Adjusted") a
   WHERE a.date >= '2021-12-09') z
   GROUP BY z.ticker;
```

date	ticker	new_return_12M	new_return_18M	new_return_24M
2022-12-09	BCD	0.17487917506003123	0.2544882413893595	0.4601116894183653
2022-12-09	BIL	0.011530904458340454	0.011093547401315284	0.01076574067926539
2022-12-09	FGINX	0.06515434257839148	0.07854907364202494	0.23077852770477497
2022-12-09	HFCVX	0.021772303811624646	0.04580191141471556	0.2417118232876759
2022-12-09	SVAAX	0.05353840720620112	0.0643415468492381	0.20656000575083233
2022-12-09	UNL	0.5730736701905823	0.8415296629300482	0.9839611064955202
2022-12-09	VNQ	-0.2324121505082119	-0.1437376333367917	0.06804493709238682

ticker	new_mu	new_sigma
BCD	0.28311824927115053	0.09676063082358806
BIL	0.001903027399710003	0.003537078726836019
FGINX	0.07030855154975779	0.07762357118705979
HFCVX	0.13182574984006268	0.07762492657565045
SVAAX	0.1277693885906032	0.05771223314533619
UNL	0.6720283260635826	0.23264176975732148
VNQ	0.005248081318067928	0.18831838612752144

```
#Portfolio's Return
SELECT a.date,
        SUM(LN(a.value / a.price_12M)*(b.quantity)) as return_new_portfolio_12M,
        SUM(LN(a.value / a.price_18M)*(b.quantity)) as return_new_portfolio_18M,
        SUM(LN(a.value / a.price_24M)*(b.quantity)) as return_new_portfolio_24M
FROM
(SELECT *,
    LAG(value, 250)
    OVER(PARTITION BY ticker
    ORDER BY date)
        AS price_12M,
    LAG(value, 375)
    OVER(PARTITION BY ticker
    ORDER BY date)
        AS price_18M,
    LAG(value, 500)
    OVER(PARTITION BY ticker
    ORDER BY date)
        AS price_24M
   FROM mytable
   WHERE price_type = "Adjusted") a
   left join holdings_new as b
   on a.ticker=b.ticker
   WHERE date = '2022-12-09';
```

```
SELECT sum(sigma*(b.quantity)) as sigma_portfolio,
SUM(risk_adj_return*b.quantity) as risk_adj_return_portfolio
FROM (
SELECT z.ticker,
    AVG(z.continuous_return) AS mu,
    STD(z.continuous_return) AS sigma
FROM
(SELECT a.*,
        LN(a.value / a.lagged_price) AS continuous_return
FROM
(SELECT *,
    LAG(value, 250)
    OVER(PARTITION BY ticker
    ORDER BY date)
        AS lagged_price
   FROM mytable
   WHERE price_type = "Adjusted") a
   WHERE a.date >= '2021-12-09') z
   GROUP BY z.ticker) s
   left join holdings_new as b
   on s.ticker=b.ticker;
```

If we take a look at the new portfolio returns, we can see that they are not only all positive, but have a way higher return compared with the previous one (going from a return -0.1859 to 0.1872 in the most recent 12 months, from -0.12 to 0.2769 in the most recent 18 months, and -0.07 to 0.408 in the past year).

date	return_new_portfolio_1...	return_new_portfolio_1...	return_new_portfolio_2...
2022-12-09	0.1871922163461774	0.27696326105990904	0.40818209901063207

Usually holdings with higher returns present higher risk. However we are getting a way higher return than before, with a portfolio with a higher weight on non-constant securities, and we obtained a lower portfolio risk, going from 0.1173 to 0.1054 with the new securities.
An explanation for that could be that our new portfolio is more diverse and diversity reduces risk.

sigma_portfolio
0.10538460320070413

References

7 Best Real Estate ETFs Of 2022 – Forbes Advisor. (n.d.). Www.forbes.com. https://www.forbes.com/advisor/investing/best-real-estate-etf/

Royal, J. (n.d.). Best Mutual Funds In November 2021. Bankrate. https://www.bankrate.com/investing/best-mutual-funds/

Top 10 Best Index Funds To Buy & Hold In 2022. (2022, November 2). FortuneBuilders. https://www.fortunebuilders.com/best-index-funds/

Total equity return REITs U.S. 2020. (n.d.). Statista. https://www.statista.com/statistics/1200045/total-return-equity-real-estate-investment-trust-reits-usa/

https://money.usnews.com/investing/funds/slideshows/best-fixed-income-funds-to-buy

https://money.usnews.com/investing/slideshows/best-reits-to-buy?slide=2

https://money.usnews.com/investing/bonds/slideshows/the-best-bond-etfs-to-buy-now?slide=14