P2 Assignment: Analytic SQL

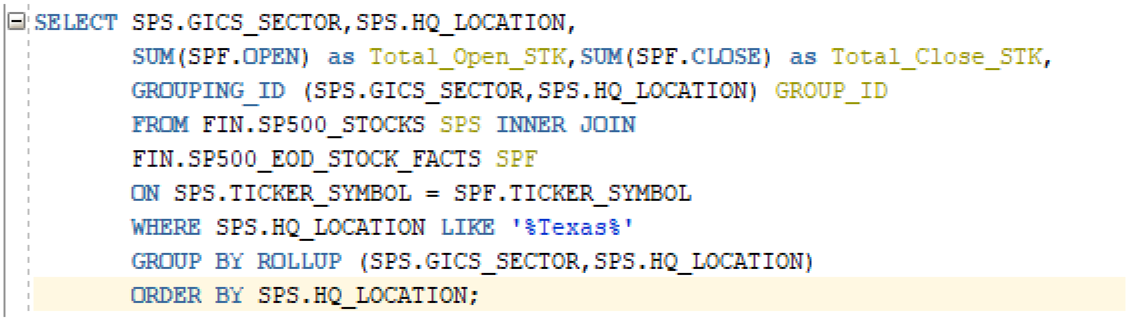
By:

Nivedita Mangal

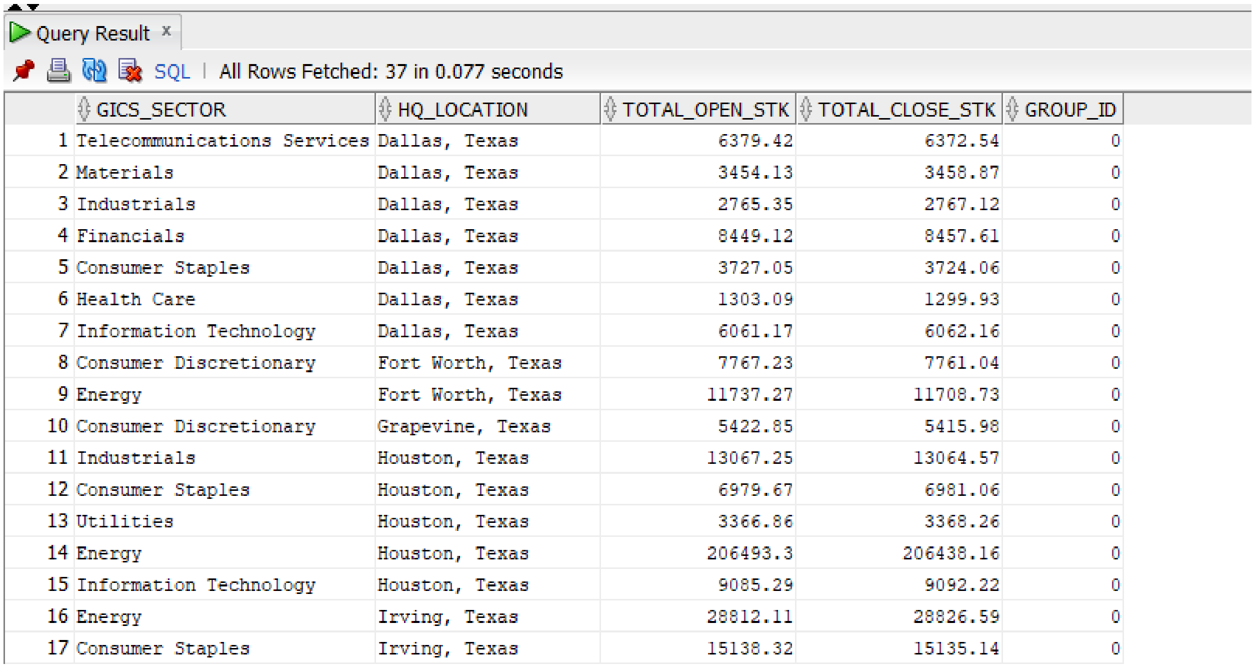
Harika Gillela

Query 1: Aggregations with CUBE and ROLLUP

Rollup:



```
SELECT SPS.GICS_SECTOR,SPS.HQ_LOCATION,
       SUM(SPF.OPEN) as Total_Open_STK,SUM(SPF.CLOSE) as Total_Close_STK,
       GROUPING_ID (SPS.GICS_SECTOR,SPS.HQ_LOCATION) GROUP_ID
       FROM FIN.SP500_STOCKS SPS INNER JOIN
       FIN.SP500_EOD_STOCK_FACTS SPF
       ON SPS.TICKER_SYMBOL = SPF.TICKER_SYMBOL
       WHERE SPS.HQ_LOCATION LIKE '%Texas%'
       GROUP BY ROLLUP (SPS.GICS_SECTOR,SPS.HQ_LOCATION)
       ORDER BY SPS.HQ_LOCATION;
```

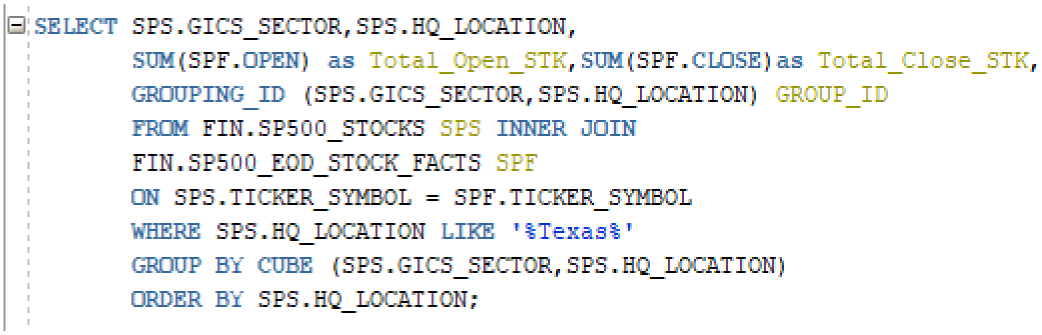


SQL | All Rows Fetched: 37 in 0.077 seconds

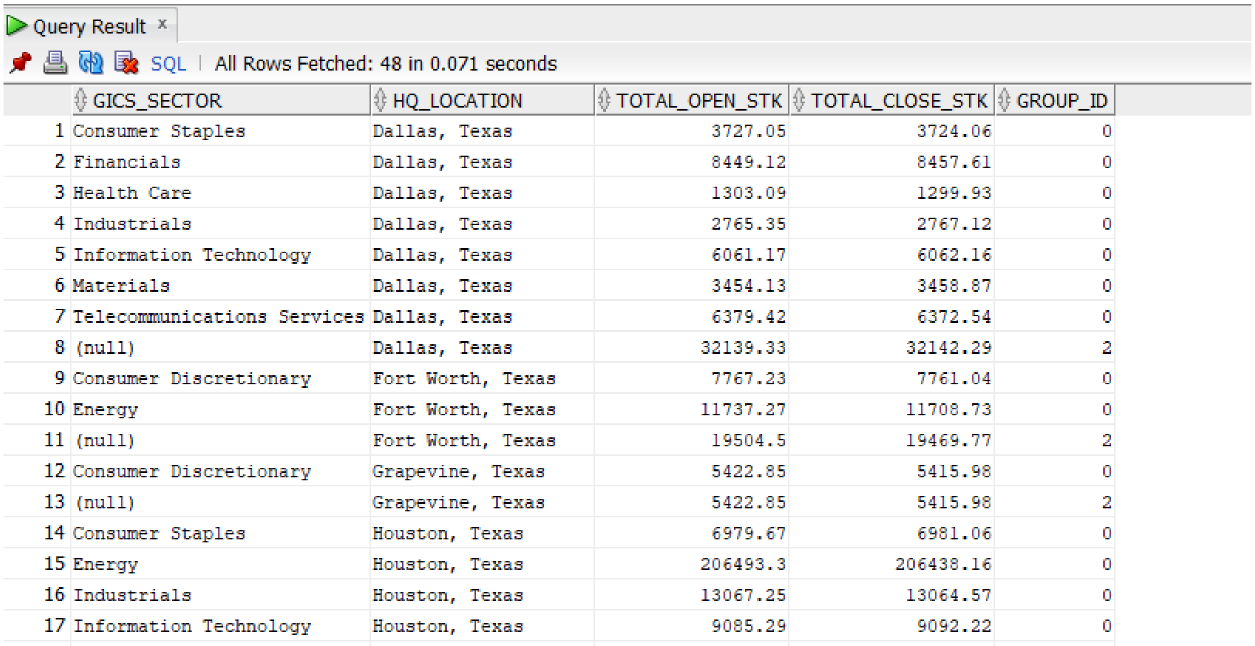
	GICS_SECTOR	HQ_LOCATION	TOTAL_OPEN_STK	TOTAL_CLOSE_STK	GROUP_ID
1	Telecommunications Services	Dallas, Texas	6379.42	6372.54	0
2	Materials	Dallas, Texas	3454.13	3458.87	0
3	Industrials	Dallas, Texas	2765.35	2767.12	0
4	Financials	Dallas, Texas	8449.12	8457.61	0
5	Consumer Staples	Dallas, Texas	3727.05	3724.06	0
6	Health Care	Dallas, Texas	1303.09	1299.93	0
7	Information Technology	Dallas, Texas	6061.17	6062.16	0
8	Consumer Discretionary	Fort Worth, Texas	7767.23	7761.04	0
9	Energy	Fort Worth, Texas	11737.27	11708.73	0
10	Consumer Discretionary	Grapevine, Texas	5422.85	5415.98	0
11	Industrials	Houston, Texas	13067.25	13064.57	0
12	Consumer Staples	Houston, Texas	6979.67	6981.06	0
13	Utilities	Houston, Texas	3366.86	3368.26	0
14	Energy	Houston, Texas	206493.3	206438.16	0
15	Information Technology	Houston, Texas	9085.29	9092.22	0
16	Energy	Irving, Texas	28812.11	28826.59	0
17	Consumer Staples	Irving, Texas	15138.32	15135.14	0

/*ROLLUP goes from left to right and CUBE takes every possible sum*/

Cube:



```
SELECT SPS.GICS_SECTOR,SPS.HQ_LOCATION,
       SUM(SPF.OPEN) as Total_Open_STK,SUM(SPF.CLOSE)as Total_Close_STK,
       GROUPING_ID (SPS.GICS_SECTOR,SPS.HQ_LOCATION) GROUP_ID
       FROM FIN.SP500_STOCKS SPS INNER JOIN
       FIN.SP500_EOD_STOCK_FACTS SPF
       ON SPS.TICKER_SYMBOL = SPF.TICKER_SYMBOL
       WHERE SPS.HQ_LOCATION LIKE '%Texas%'
       GROUP BY CUBE (SPS.GICS_SECTOR,SPS.HQ_LOCATION)
       ORDER BY SPS.HQ_LOCATION;
```

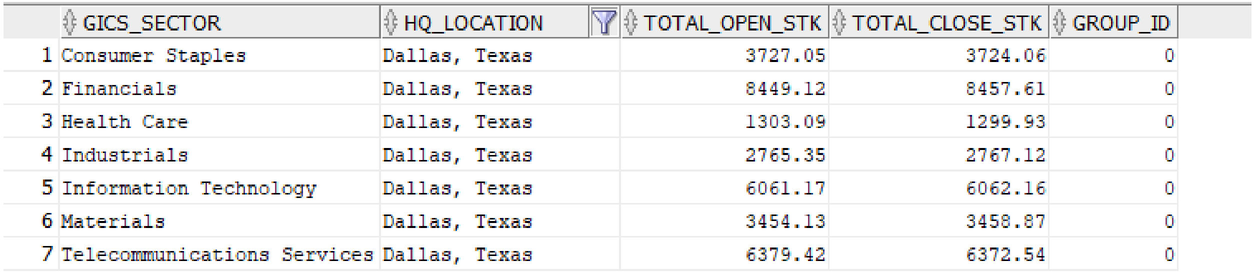


All Rows Fetched: 48 in 0.071 seconds

	GICS_SECTOR	HQ_LOCATION	TOTAL_OPEN_STK	TOTAL_CLOSE_STK	GROUP_ID
1	Consumer Staples	Dallas, Texas	3727.05	3724.06	0
2	Financials	Dallas, Texas	8449.12	8457.61	0
3	Health Care	Dallas, Texas	1303.09	1299.93	0
4	Industrials	Dallas, Texas	2765.35	2767.12	0
5	Information Technology	Dallas, Texas	6061.17	6062.16	0
6	Materials	Dallas, Texas	3454.13	3458.87	0
7	Telecommunications Services	Dallas, Texas	6379.42	6372.54	0
8	(null)	Dallas, Texas	32139.33	32142.29	2
9	Consumer Discretionary	Fort Worth, Texas	7767.23	7761.04	0
10	Energy	Fort Worth, Texas	11737.27	11708.73	0
11	(null)	Fort Worth, Texas	19504.5	19469.77	2
12	Consumer Discretionary	Grapevine, Texas	5422.85	5415.98	0
13	(null)	Grapevine, Texas	5422.85	5415.98	2
14	Consumer Staples	Houston, Texas	6979.67	6981.06	0
15	Energy	Houston, Texas	206493.3	206438.16	0
16	Industrials	Houston, Texas	13067.25	13064.57	0
17	Information Technology	Houston, Texas	9085.29	9092.22	0

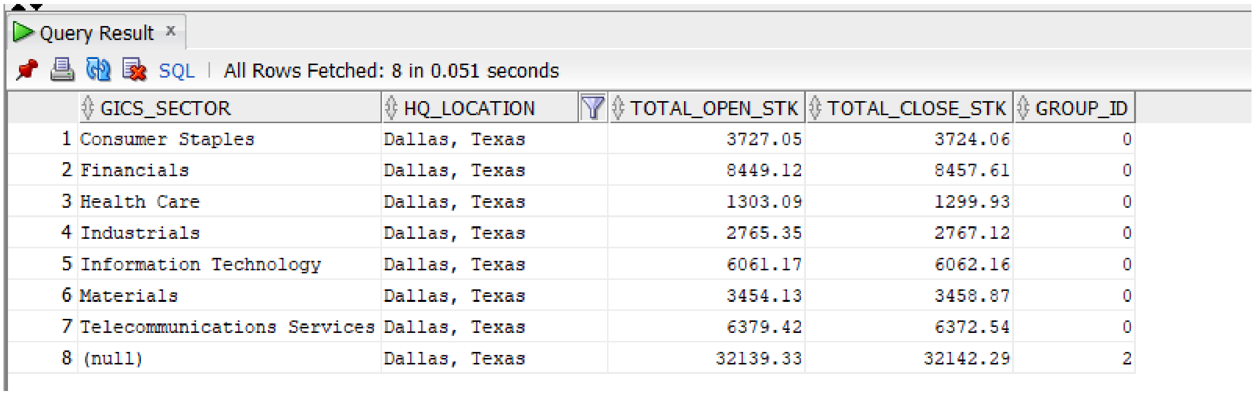
Comparison of Rollup and Cube

Rollup:



	GICS_SECTOR	HQ_LOCATION	TOTAL_OPEN_STK	TOTAL_CLOSE_STK	GROUP_ID
1	Consumer Staples	Dallas, Texas	3727.05	3724.06	0
2	Financials	Dallas, Texas	8449.12	8457.61	0
3	Health Care	Dallas, Texas	1303.09	1299.93	0
4	Industrials	Dallas, Texas	2765.35	2767.12	0
5	Information Technology	Dallas, Texas	6061.17	6062.16	0
6	Materials	Dallas, Texas	3454.13	3458.87	0
7	Telecommunications Services	Dallas, Texas	6379.42	6372.54	0

Cube:



All Rows Fetched: 8 in 0.051 seconds

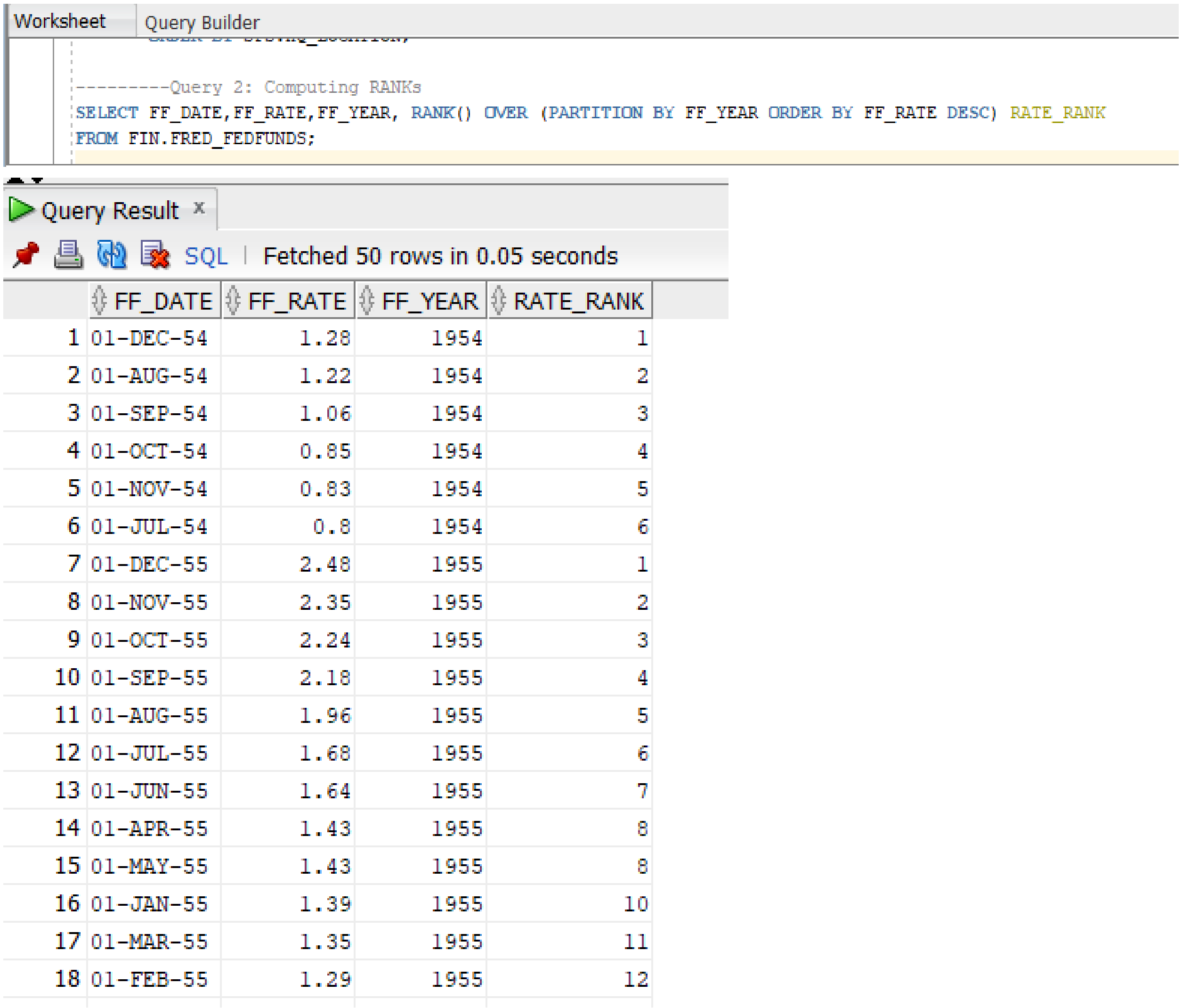
	GICS_SECTOR	HQ_LOCATION	TOTAL_OPEN_STK	TOTAL_CLOSE_STK	GROUP_ID
1	Consumer Staples	Dallas, Texas	3727.05	3724.06	0
2	Financials	Dallas, Texas	8449.12	8457.61	0
3	Health Care	Dallas, Texas	1303.09	1299.93	0
4	Industrials	Dallas, Texas	2765.35	2767.12	0
5	Information Technology	Dallas, Texas	6061.17	6062.16	0
6	Materials	Dallas, Texas	3454.13	3458.87	0
7	Telecommunications Services	Dallas, Texas	6379.42	6372.54	0
8	(null)	Dallas, Texas	32139.33	32142.29	2

In above example, ROLLUP produces only a fraction of possible subtotal combinations, whereas CUBE produces subtotals for all possible combinations of groupings specified in the GROUP BY clause, and a grand total.

In the query result with roll up function we do not have the subtotal for open and close stock prices for location. We have subtotal for each sector, sector and location and grand total. The cube function provides subtotal for sector, location, sector and location and grand total.

Query 2: Computing RANKs

Rate Rank:

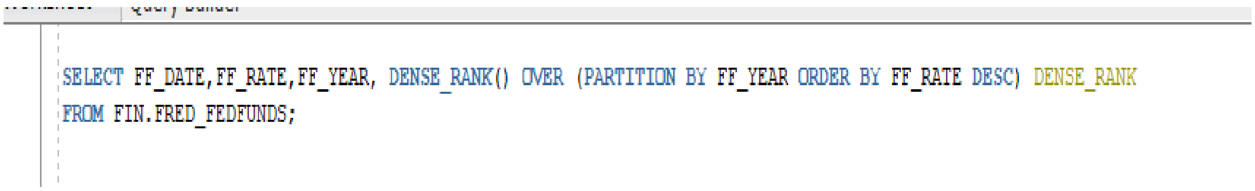


```
---------Query 2: Computing RANKs
SELECT FF_DATE,FF_RATE,FF_YEAR, RANK() OVER (PARTITION BY FF_YEAR ORDER BY FF_RATE DESC) RATE_RANK
FROM FIN.FRED_FEDFUNDS;
```

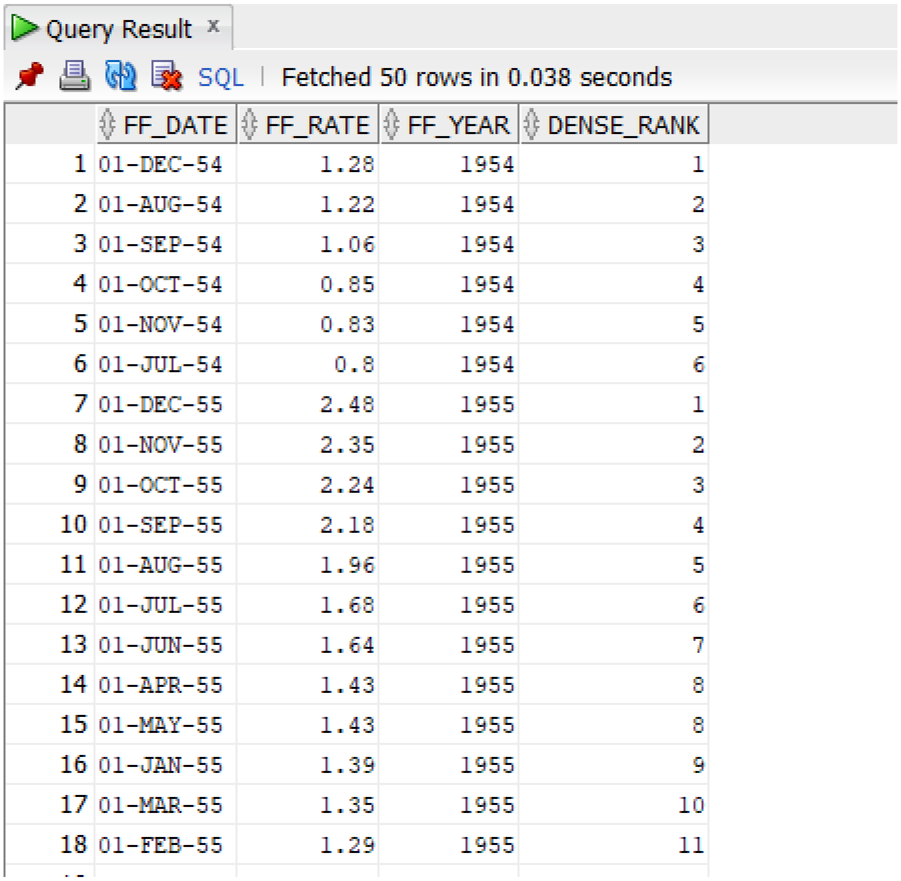
Fetched 50 rows in 0.05 seconds

	FF_DATE	FF_RATE	FF_YEAR	RATE_RANK
1	01-DEC-54	1.28	1954	1
2	01-AUG-54	1.22	1954	2
3	01-SEP-54	1.06	1954	3
4	01-OCT-54	0.85	1954	4
5	01-NOV-54	0.83	1954	5
6	01-JUL-54	0.8	1954	6
7	01-DEC-55	2.48	1955	1
8	01-NOV-55	2.35	1955	2
9	01-OCT-55	2.24	1955	3
10	01-SEP-55	2.18	1955	4
11	01-AUG-55	1.96	1955	5
12	01-JUL-55	1.68	1955	6
13	01-JUN-55	1.64	1955	7
14	01-APR-55	1.43	1955	8
15	01-MAY-55	1.43	1955	8
16	01-JAN-55	1.39	1955	10
17	01-MAR-55	1.35	1955	11
18	01-FEB-55	1.29	1955	12

Dense_Rank:



```
SELECT FF_DATE,FF_RATE,FF_YEAR, DENSE_RANK() OVER (PARTITION BY FF_YEAR ORDER BY FF_RATE DESC) DENSE_RANK
FROM FIN.FRED_FEDFUNDS;
```

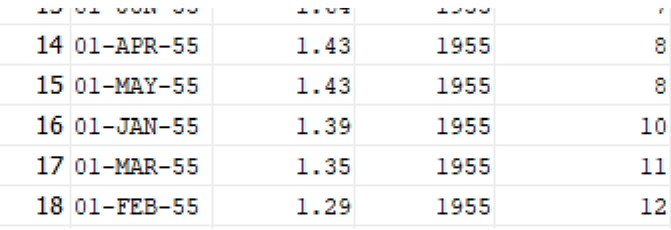


Fetched 50 rows in 0.038 seconds

	FF_DATE	FF_RATE	FF_YEAR	DENSE_RANK
1	01-DEC-54	1.28	1954	1
2	01-AUG-54	1.22	1954	2
3	01-SEP-54	1.06	1954	3
4	01-OCT-54	0.85	1954	4
5	01-NOV-54	0.83	1954	5
6	01-JUL-54	0.8	1954	6
7	01-DEC-55	2.48	1955	1
8	01-NOV-55	2.35	1955	2
9	01-OCT-55	2.24	1955	3
10	01-SEP-55	2.18	1955	4
11	01-AUG-55	1.96	1955	5
12	01-JUL-55	1.68	1955	6
13	01-JUN-55	1.64	1955	7
14	01-APR-55	1.43	1955	8
15	01-MAY-55	1.43	1955	8
16	01-JAN-55	1.39	1955	9
17	01-MAR-55	1.35	1955	10
18	01-FEB-55	1.29	1955	11

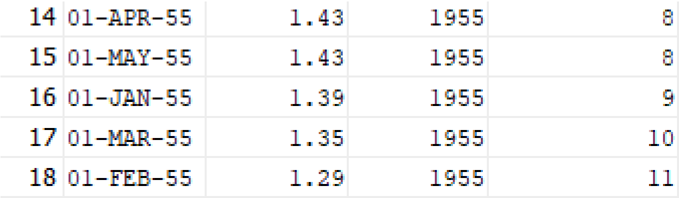
Comparing Rate and Dense rank:

Rate rank:



13	01-JUN-55	1.64	1955	7
14	01-APR-55	1.43	1955	8
15	01-MAY-55	1.43	1955	8
16	01-JAN-55	1.39	1955	10
17	01-MAR-55	1.35	1955	11
18	01-FEB-55	1.29	1955	12

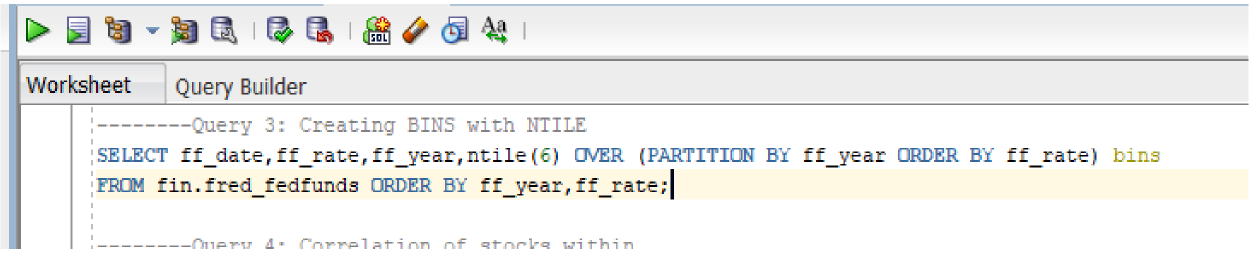
Dense rank:



14	01-APR-55	1.43	1955	8
15	01-MAY-55	1.43	1955	8
16	01-JAN-55	1.39	1955	9
17	01-MAR-55	1.35	1955	10
18	01-FEB-55	1.29	1955	11

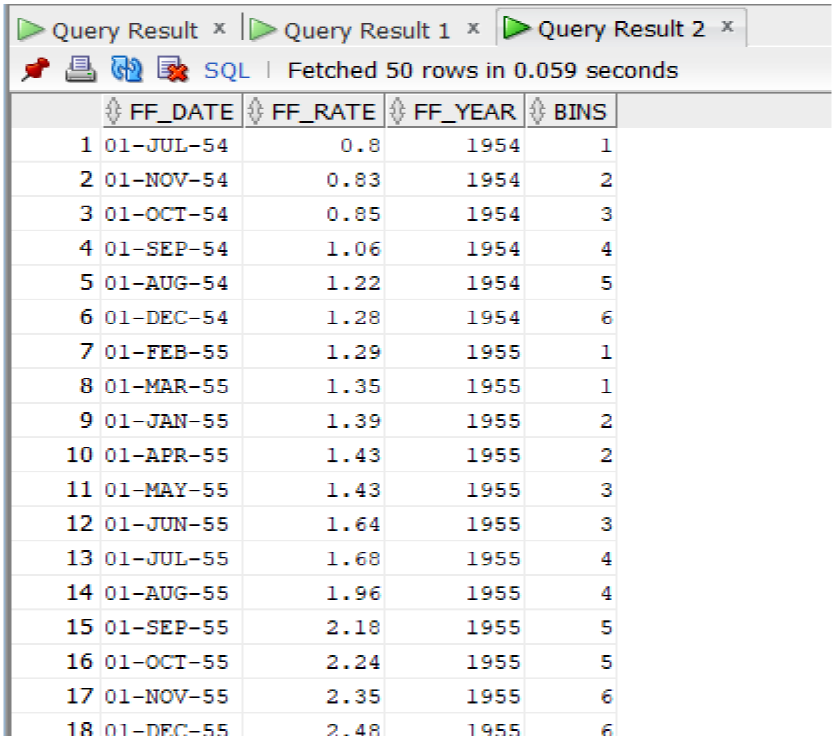
In RATE RANK, all the rows with same FF_RATE will have same rank and subsequent rank is skipped and will not be in a sequence. To avoid this issue, we use DENSE RANK. Dense will not skip the numbers when we have rows with the same values.

Query 3: Creating Bins with NTILE



```
--------Query 3: Creating BINS with NTILE
SELECT ff_date,ff_rate,ff_year,ntile(6) OVER (PARTITION BY ff_year ORDER BY ff_rate) bins
FROM fin.fred_fedfunds ORDER BY ff_year,ff_rate;

--------Query 4: Correlation of stocks within
```



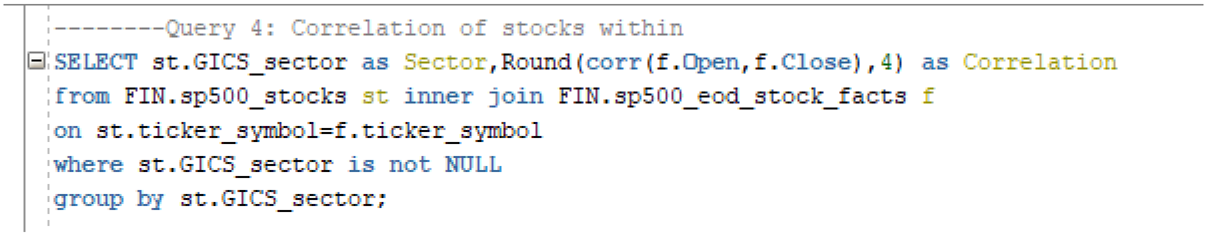
Fetched 50 rows in 0.059 seconds

	FF_DATE	FF_RATE	FF_YEAR	BINS
1	01-JUL-54	0.8	1954	1
2	01-NOV-54	0.83	1954	2
3	01-OCT-54	0.85	1954	3
4	01-SEP-54	1.06	1954	4
5	01-AUG-54	1.22	1954	5
6	01-DEC-54	1.28	1954	6
7	01-FEB-55	1.29	1955	1
8	01-MAR-55	1.35	1955	1
9	01-JAN-55	1.39	1955	2
10	01-APR-55	1.43	1955	2
11	01-MAY-55	1.43	1955	3
12	01-JUN-55	1.64	1955	3
13	01-JUL-55	1.68	1955	4
14	01-AUG-55	1.96	1955	4
15	01-SEP-55	2.18	1955	5
16	01-OCT-55	2.24	1955	5
17	01-NOV-55	2.35	1955	6
18	01-DEC-55	2.48	1955	6

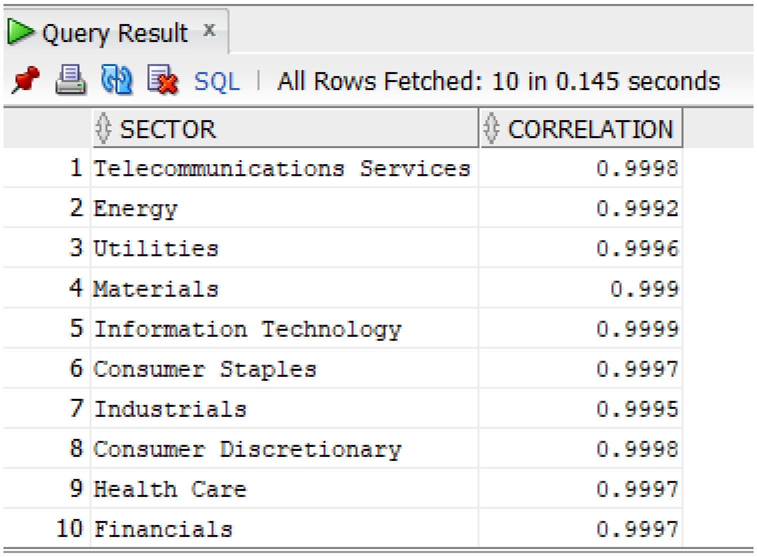
The NTILE analytic function allows you to break a result set into a specified number of approximately equal groups. Here we have written a query that divides the data in to 6 bins. The partition by clause divides the result set in to bins for each year based on the order of FF_RATE.

Query 4: Correlations (CORR)

Correlation of stocks within sector:



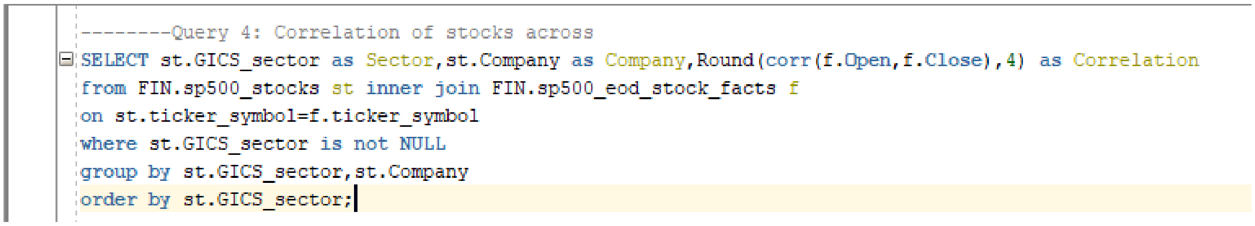
```
--------Query 4: Correlation of stocks within
SELECT st.GICS_sector as Sector,Round(corr(f.Open,f.Close),4) as Correlation
from FIN.sp500_stocks st inner join FIN.sp500_eod_stock_facts f
on st.ticker_symbol=f.ticker_symbol
where st.GICS_sector is not NULL
group by st.GICS_sector;
```



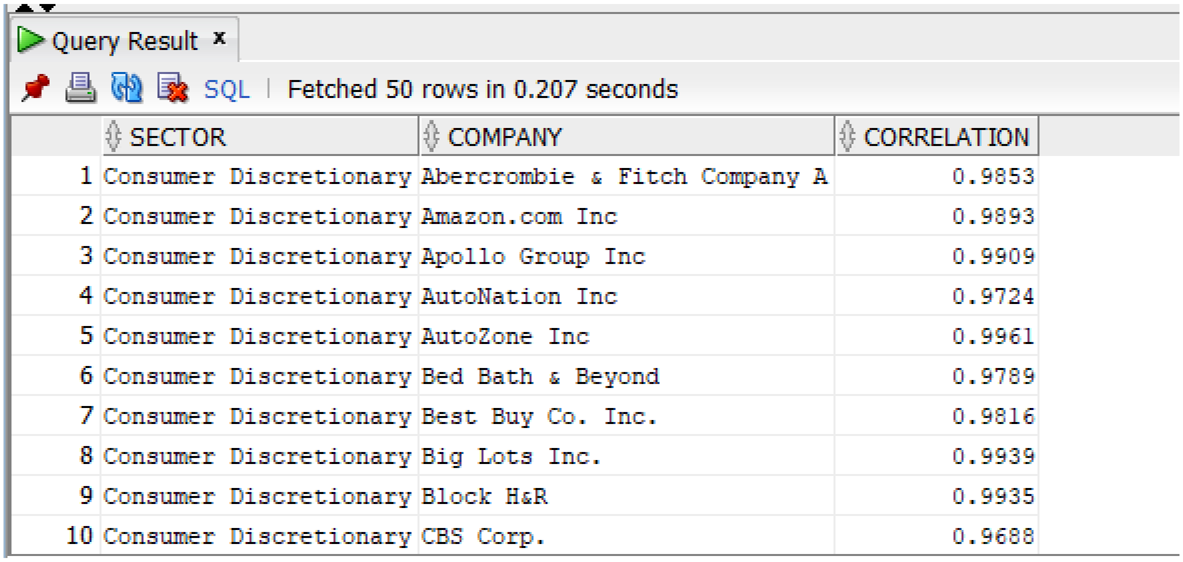
All Rows Fetched: 10 in 0.145 seconds

	SECTOR	CORRELATION
1	Telecommunications Services	0.9998
2	Energy	0.9992
3	Utilities	0.9996
4	Materials	0.999
5	Information Technology	0.9999
6	Consumer Staples	0.9997
7	Industrials	0.9995
8	Consumer Discretionary	0.9998
9	Health Care	0.9997
10	Financials	0.9997

Correlation of stocks across sector:



```
--------Query 4: Correlation of stocks across
SELECT st.GICS_sector as Sector,st.Company as Company,Round(corr(f.Open,f.Close),4) as Correlation
from FIN.sp500_stocks st inner join FIN.sp500_eod_stock_facts f
on st.ticker_symbol=f.ticker_symbol
where st.GICS_sector is not NULL
group by st.GICS_sector,st.Company
order by st.GICS_sector;
```

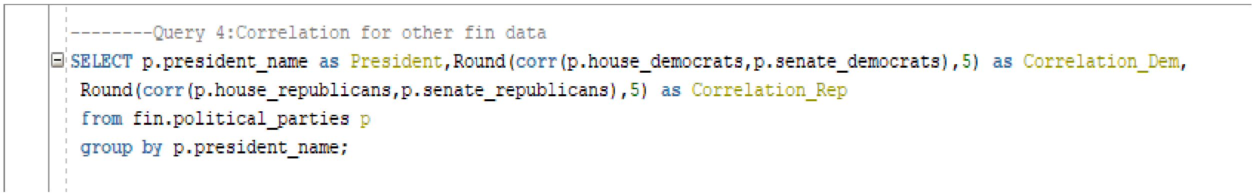


Fetched 50 rows in 0.207 seconds

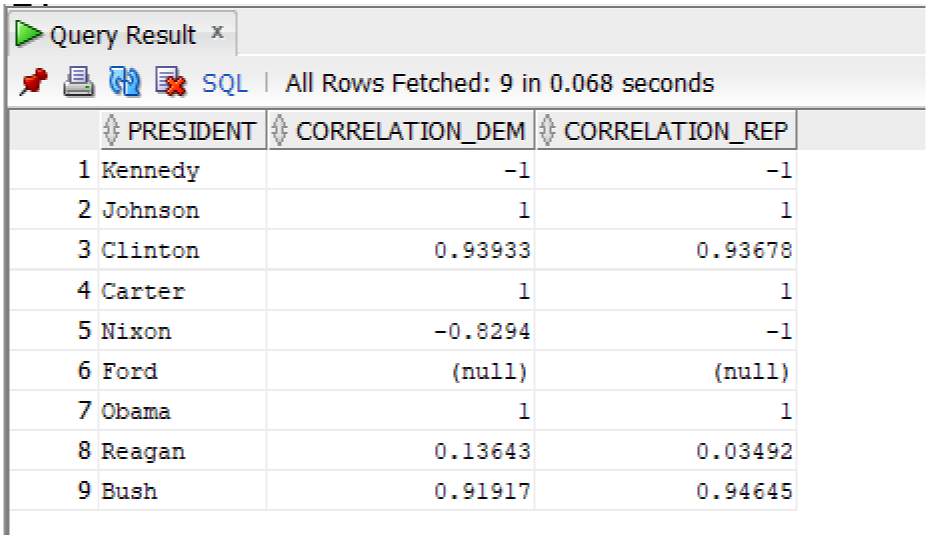
	SECTOR	COMPANY	CORRELATION
1	Consumer Discretionary	Abercrombie & Fitch Company A	0.9853
2	Consumer Discretionary	Amazon.com Inc	0.9893
3	Consumer Discretionary	Apollo Group Inc	0.9909
4	Consumer Discretionary	AutoNation Inc	0.9724
5	Consumer Discretionary	AutoZone Inc	0.9961
6	Consumer Discretionary	Bed Bath & Beyond	0.9789
7	Consumer Discretionary	Best Buy Co. Inc.	0.9816
8	Consumer Discretionary	Big Lots Inc.	0.9939
9	Consumer Discretionary	Block H&R	0.9935
10	Consumer Discretionary	CBS Corp.	0.9688

CORR returns the coefficient of correlation of a set of number pairs. Here we are trying to find the correlation between the open and close prices of the stocks for each sector.
Further we have also checked the correlation between open and close stock prices among various companies of various sector.

Correlation for other finance data:



```
--------Query 4:Correlation for other fin data
SELECT p.president_name as President,Round(corr(p.house_democrats,p.senate_democrats),5) as Correlation_Dem,
 Round(corr(p.house_republicans,p.senate_republicans),5) as Correlation_Rep
 from fin.political_parties p
 group by p.president_name;
```



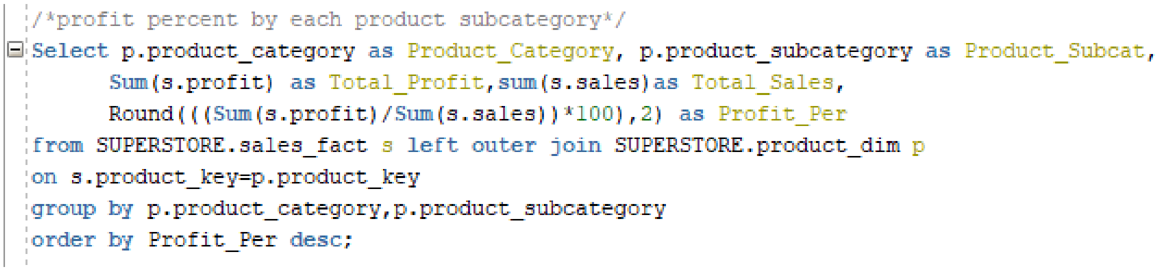
All Rows Fetched: 9 in 0.068 seconds

	PRESIDENT	CORRELATION_DEM	CORRELATION_REP
1	Kennedy	-1	-1
2	Johnson	1	1
3	Clinton	0.93933	0.93678
4	Carter	1	1
5	Nixon	-0.8294	-1
6	Ford	(null)	(null)
7	Obama	1	1
8	Reagan	0.13643	0.03492
9	Bush	0.91917	0.94645

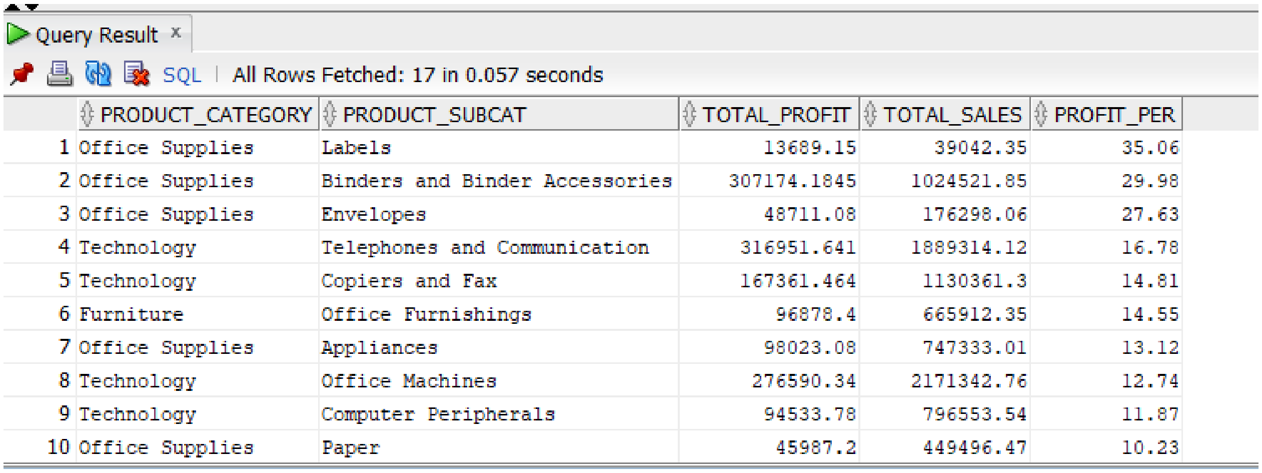
Additionally, we tried to check the correlation between house democrat, senate democrat and house republican and senate republican for each president of United States.

Query 5: Interesting Queries

Query 1:



```
/*profit percent by each product subcategory*/
Select p.product_category as Product_Category, p.product_subcategory as Product_Subcat,
      Sum(s.profit) as Total_Profit,sum(s.sales)as Total_Sales,
      Round(((Sum(s.profit)/Sum(s.sales))*100),2) as Profit_Per
from SUPERSTORE.sales_fact s left outer join SUPERSTORE.product_dim p
on s.product_key=p.product_key
group by p.product_category,p.product_subcategory
order by Profit_Per desc;
```

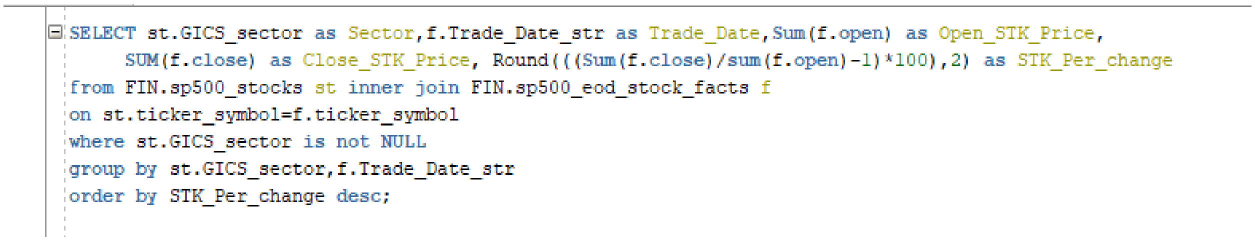


All Rows Fetched: 17 in 0.057 seconds

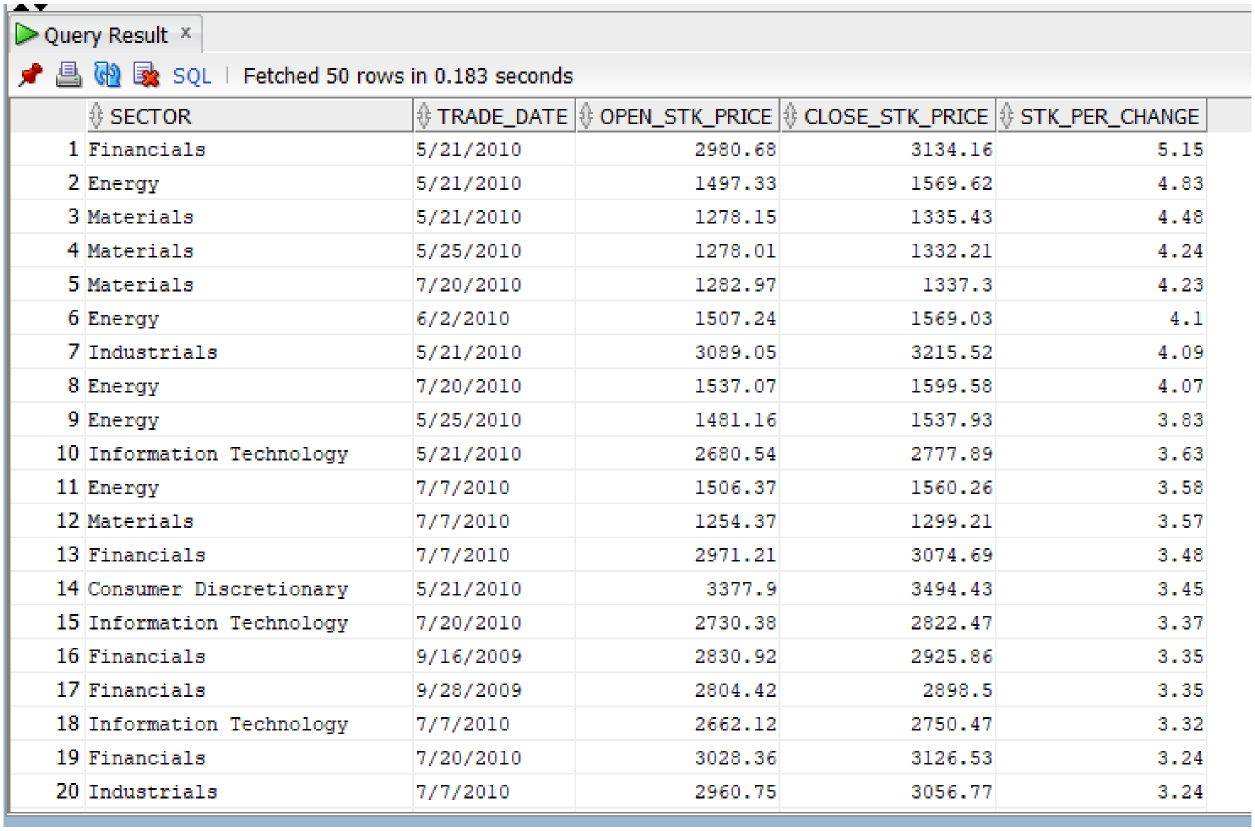
	PRODUCT_CATEGORY	PRODUCT_SUBCAT	TOTAL_PROFIT	TOTAL_SALES	PROFIT_PER
1	Office Supplies	Labels	13689.15	39042.35	35.06
2	Office Supplies	Binders and Binder Accessories	307174.1845	1024521.85	29.98
3	Office Supplies	Envelopes	48711.08	176298.06	27.63
4	Technology	Telephones and Communication	316951.641	1889314.12	16.78
5	Technology	Copiers and Fax	167361.464	1130361.3	14.81
6	Furniture	Office Furnishings	96878.4	665912.35	14.55
7	Office Supplies	Appliances	98023.08	747333.01	13.12
8	Technology	Office Machines	276590.34	2171342.76	12.74
9	Technology	Computer Peripherals	94533.78	796553.54	11.87
10	Office Supplies	Paper	45987.2	449496.47	10.23

This query calculates the, total profit, total sales and profit percentage from the superstores data for each product category and sub-category and display the product category and sub-category from highest to lowest profit percentage.

Query 2:



```
SELECT st.GICS_sector as Sector,f.Trade_Date_str as Trade_Date,Sum(f.open) as Open_STK_Price,
     SUM(f.close) as Close_STK_Price, Round(((Sum(f.close)/sum(f.open)-1)*100),2) as STK_Per_change
from FIN.sp500_stocks st inner join FIN.sp500_eod_stock_facts f
on st.ticker_symbol=f.ticker_symbol
where st.GICS_sector is not NULL
group by st.GICS_sector,f.Trade_Date_str
order by STK_Per_change desc;
```

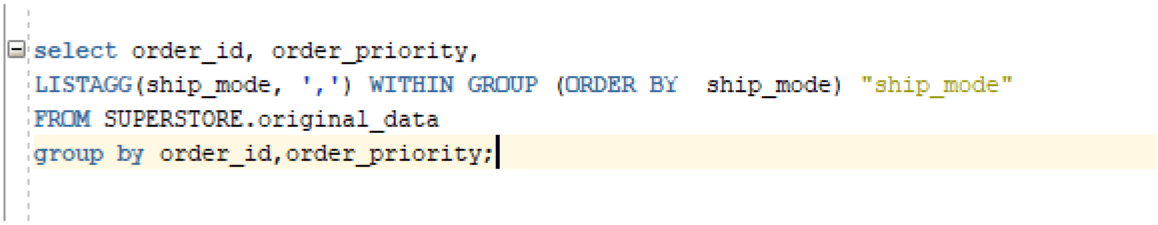


Fetched 50 rows in 0.183 seconds

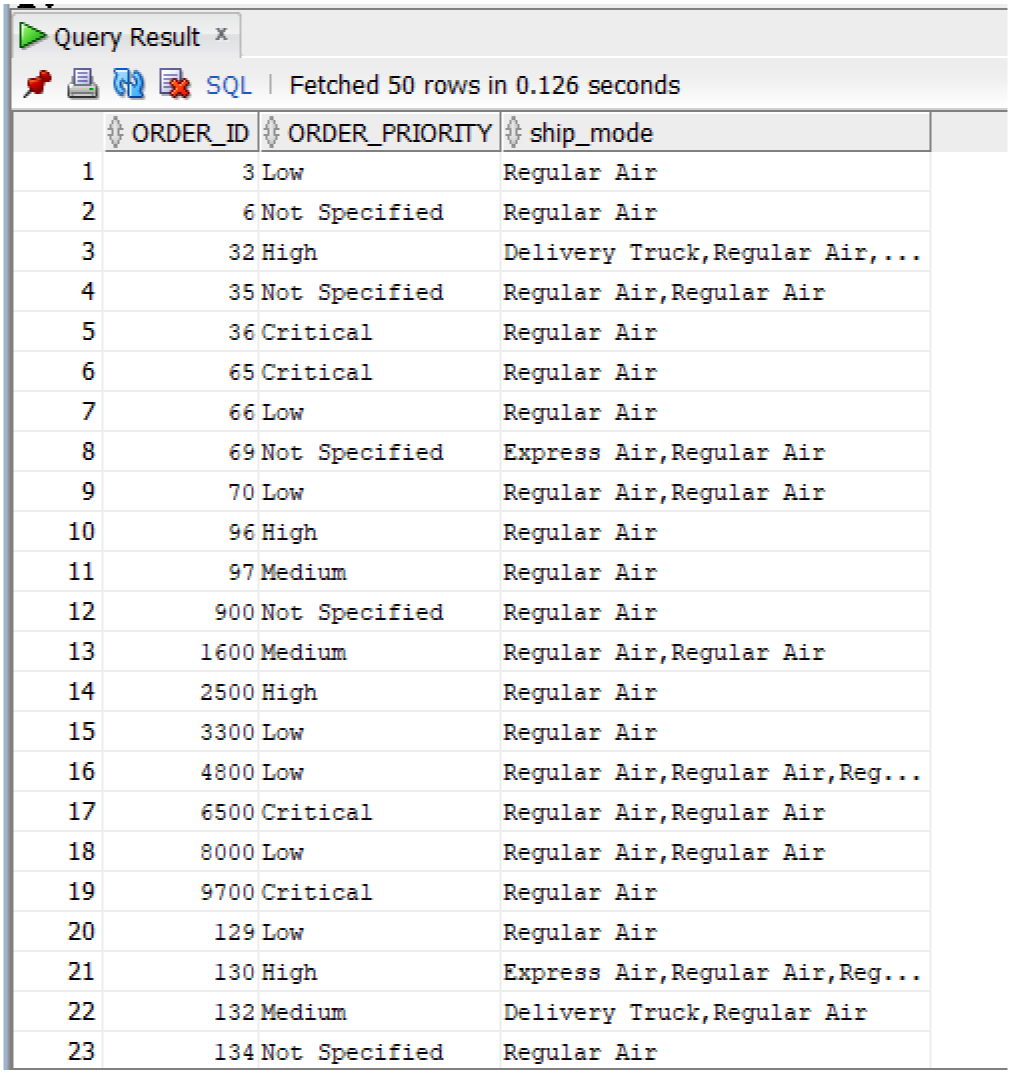
	SECTOR	TRADE_DATE	OPEN_STK_PRICE	CLOSE_STK_PRICE	STK_PER_CHANGE
1	Financials	5/21/2010	2980.68	3134.16	5.15
2	Energy	5/21/2010	1497.33	1569.62	4.83
3	Materials	5/21/2010	1278.15	1335.43	4.48
4	Materials	5/25/2010	1278.01	1332.21	4.24
5	Materials	7/20/2010	1282.97	1337.3	4.23
6	Energy	6/2/2010	1507.24	1569.03	4.1
7	Industrials	5/21/2010	3089.05	3215.52	4.09
8	Energy	7/20/2010	1537.07	1599.58	4.07
9	Energy	5/25/2010	1481.16	1537.93	3.83
10	Information Technology	5/21/2010	2680.54	2777.89	3.63
11	Energy	7/7/2010	1506.37	1560.26	3.58
12	Materials	7/7/2010	1254.37	1299.21	3.57
13	Financials	7/7/2010	2971.21	3074.69	3.48
14	Consumer Discretionary	5/21/2010	3377.9	3494.43	3.45
15	Information Technology	7/20/2010	2730.38	2822.47	3.37
16	Financials	9/16/2009	2830.92	2925.86	3.35
17	Financials	9/28/2009	2804.42	2898.5	3.35
18	Information Technology	7/7/2010	2662.12	2750.47	3.32
19	Financials	7/20/2010	3028.36	3126.53	3.24
20	Industrials	7/7/2010	2960.75	3056.77	3.24

This query will calculate the percentage change in the stock price for each day and for each sector with the highest percentage change to the lowest percentage change. The value is calculated based on the open and close stock prices.

Query 3:



```
select order_id, order_priority,
LISTAGG(ship_mode, ',') WITHIN GROUP (ORDER BY  ship_mode) "ship_mode"
FROM SUPERSTORE.original_data
group by order_id,order_priority;
```



Fetched 50 rows in 0.126 seconds

	ORDER_ID	ORDER_PRIORITY	ship_mode
1	3	Low	Regular Air
2	6	Not Specified	Regular Air
3	32	High	Delivery Truck,Regular Air,...
4	35	Not Specified	Regular Air,Regular Air
5	36	Critical	Regular Air
6	65	Critical	Regular Air
7	66	Low	Regular Air
8	69	Not Specified	Express Air,Regular Air
9	70	Low	Regular Air,Regular Air
10	96	High	Regular Air
11	97	Medium	Regular Air
12	900	Not Specified	Regular Air
13	1600	Medium	Regular Air,Regular Air
14	2500	High	Regular Air
15	3300	Low	Regular Air
16	4800	Low	Regular Air,Regular Air,Reg...
17	6500	Critical	Regular Air,Regular Air
18	8000	Low	Regular Air,Regular Air
19	9700	Critical	Regular Air
20	129	Low	Regular Air
21	130	High	Express Air,Regular Air,Reg...
22	132	Medium	Delivery Truck,Regular Air
23	134	Not Specified	Regular Air

LISTAGG orders data within each group specified in the ORDER BY clause and then concatenates the values of the measure column. Here we have concatenated the shipping modes for multiple products under each order and the corresponding order priority.