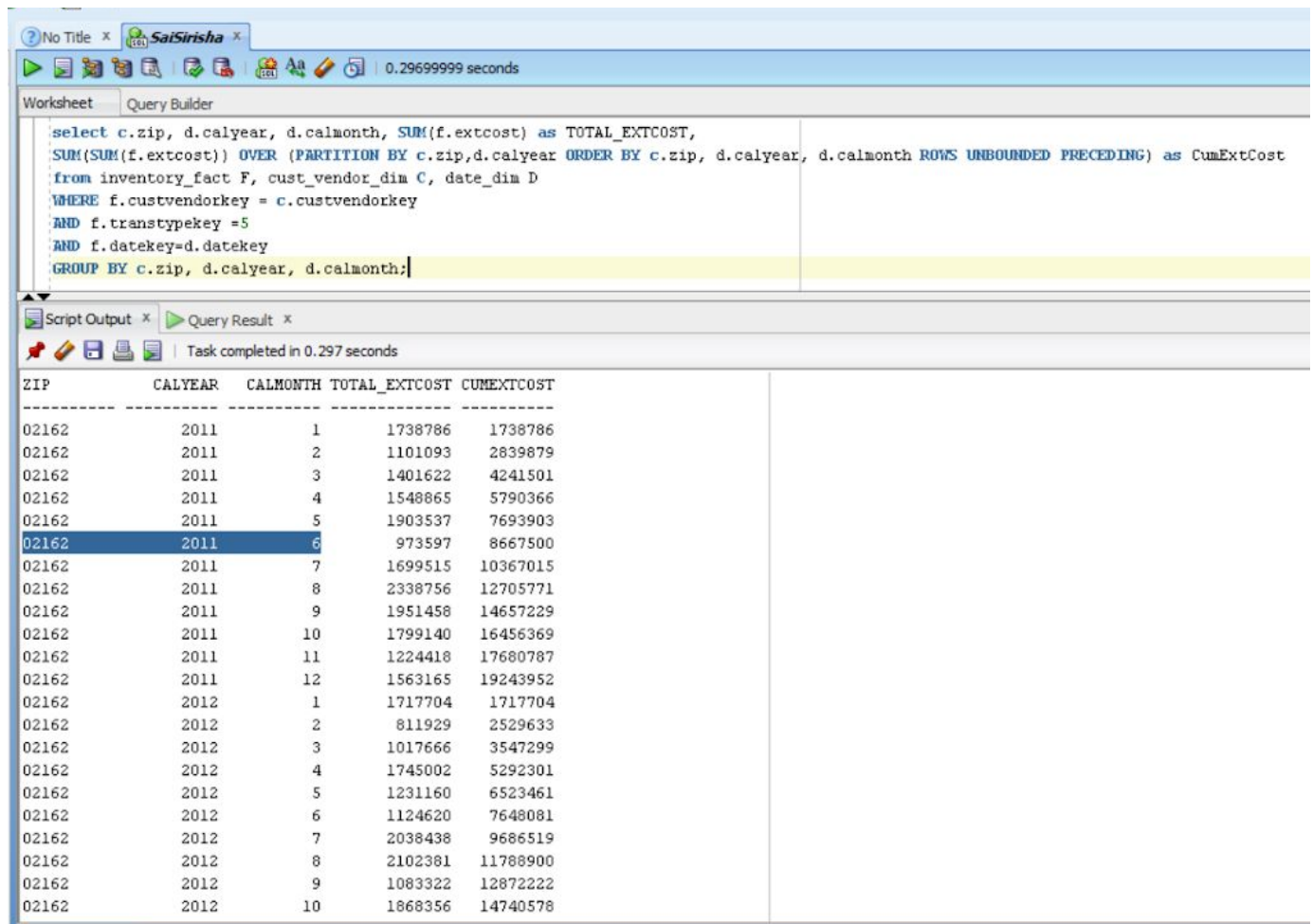


QUERY 5

```
select c.zip, d.calyear, d.calmonth, SUM(f.extcost) as TOTAL_EXTCOST,  
SUM(SUM(f.extcost)) OVER (PARTITION BY c.zip,d.calyear ORDER BY c.zip, d.calyear,  
d.calmonth ROWS UNBOUNDED PRECEDING) as CumExtCost  
from inventory_fact F, cust_vendor_dim C, date_dim D  
WHERE f.custvendorkey = c.custvendorkey  
AND f.transtypekey =5  
AND f.datekey=d.datekey  
GROUP BY c.zip, d.calyear, d.calmonth;
```



The screenshot displays the SaiSirisha Query Builder interface. The top toolbar includes icons for running the query, saving, and other functions, along with a timer showing 0.29699999 seconds. The 'Query Builder' tab is active, showing the SQL query in the editor. Below the editor, the 'Script Output' and 'Query Result' tabs are visible. The 'Query Result' tab shows the output of the query, which is a table with 5 columns: ZIP, CALYEAR, CALMONTH, TOTAL_EXTCOST, and CUMEXTCOST. The table contains 20 rows of data, representing the cumulative cost for each month from January to December for the year 2011 and 2012, grouped by ZIP code 02162.

ZIP	CALYEAR	CALMONTH	TOTAL_EXTCOST	CUMEXTCOST
02162	2011	1	1738786	1738786
02162	2011	2	1101093	2839879
02162	2011	3	1401622	4241501
02162	2011	4	1548865	5790366
02162	2011	5	1903537	7693903
02162	2011	6	973597	8667500
02162	2011	7	1699515	10367015
02162	2011	8	2338756	12705771
02162	2011	9	1951458	14657229
02162	2011	10	1799140	16456369
02162	2011	11	1224418	17680787
02162	2011	12	1563165	19243952
02162	2012	1	1717704	1717704
02162	2012	2	811929	2529633
02162	2012	3	1017666	3547299
02162	2012	4	1745002	5292301
02162	2012	5	1231160	6523461
02162	2012	6	1124620	7648081
02162	2012	7	2038438	9686519
02162	2012	8	2102381	11788900
02162	2012	9	1083322	12872222
02162	2012	10	1868356	14740578